

Отчет по лабораторной работе №10

Простейший вариант

Шубина София Антоновна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	13
	Список литературы	14

Список иллюстраций

2.1	Создание файлов и каталогов	6
2.2	Написание программы	7
2.3	Создаем исполняемый файл и проверяем его работу	7
2.4	Выполнение файла с измененным првом доступа	8
2.5	Изменение прав доступа и запуск файла	8
2.6	Предоставление прав доступа к файлу и проверка правильности выполнения	9
2.7	Написание программы	10
2.8	Проверка наличия файла и его содержимого с помощью команд ls,cat	10

Список таблиц

1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Выполнение лабораторной работы

1. Создадим каталог для программ лабораторной работы No 10, перейдем в него и создадим файлы lab10-1.asm, readme-1.txt и readme-2.txt: mkdir ~/work/arch-pc/lab09 cd ~/work/arch-pc/lab09 touch lab10-1.asm readme-1.txt readme-2.txt (рис. 2.1).

```
sashubina@dk6n62 ~ $ mkdir ~/work/arch-pc/lab10
sashubina@dk6n62 ~ $ cd ~/work/arch-pc/lab10
sashubina@dk6n62 ~/work/arch-pc/lab10 $ touch lab10-1.asm readme-1.txt readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $
```

Рис. 2.1: Создание файлов и каталогов

2. Введем в файл lab10-1.asm текст программы из листинга (Программа записи в файл сообщения). Создаем исполняемый файл и проверим его работу. (рис. 2.2 , 2.3)

```

lab10-1.asm      [-M--]  9 L:[ 1+39  40/ 40] *(1286/1286b) <EOF>
;~~~~~
; Запись в файл строки введенной на запрос
;~~~~~
%include "lab10-1.asm"
SECTION .text
filename db "lab10-1.txt", 0h ; Имя файла
msg db "Введите строку для записи в файл: ", 0h ; Сообщение
SECTION .data
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

```

Рис. 2.2: Написание программы

```

sashubina@dk6n62 ~/work/arch-pc/lab10 $ nasm -f elf lab10-1.asm
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: Sofia Shubina
sashubina@dk6n62 ~/work/arch-pc/lab10 $

```

Рис. 2.3: Создаем исполняемый файл и проверяем его работу

3. С помощью команды `chmod` изменим права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. Попробуем выполнить файл. Объясним результат.-Файл не выполнился,потому что мы запретили прва доступа к выполнению (рис. 2.4).

```
sashubina@dk6n62 ~/work/arch-pc/lab10 $ chmod 666 lab10-1
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l lab10-1
-rw-rw-rw- 1 sashubina studsci 9164 ноя 20 13:50 lab10-1
sashubina@dk6n62 ~/work/arch-pc/lab10 $
```

Рис. 2.4: Выполнение файла с измененным првом доступа

4. С помощью команды `chmod` измените права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Попробуйте выполнить его и объясните результат.-Текстовый файл не должен исполняться даже после разрешения доступа к выполнению, потому что он не является исполняемым. (рис. 2.5).

```
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-1.txt
-rw-r--r-- 1 sashubina studsci 0 ноя 20 13:43 readme-1.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ chmod 317 readme-1.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-1.txt
--wx--xrw- 1 sashubina studsci 0 ноя 20 13:43 readme-1.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ chmod 252 readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-2.txt
bash: ls-l: команда не найдена
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-2.txt
--w-r-x-w- 1 sashubina studsci 0 ноя 20 13:43 readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $
```

Рис. 2.5: Изменение прав доступа и запуск файла

5. В соответствии с вариантом в таблице 10.4 предоставить права доступа к файлу `readme- 1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде. Права на доступ к файлу предоставляются в символьном виде по разрешению преподавателя,т.к. в двоичном виде программа не работает Проверить правильность выполнения с помощью команды `ls -l`. (рис. 2.6).


```

sashubina@dk6n62 ~/work/arch-pc/lab10 $ chmod 252 readme-1.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-1.txt
--w-r-x-w- 1 sashubina studsci 0 ноя 20 13:43 readme-1.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ chmod 317 readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls -l readme-2.txt
--wx--x-rwx 1 sashubina studsci 0 ноя 20 13:43 readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $

```

Рис. 2.6: Предоставление прав доступа к файлу и проверка правильности выполнения

#Задание для самостоятельной работы 1. Напишите программу работающую 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

Архитектура ЭВМ Создать исполняемый файл и проверить его работу. Проверить наличие файла и его содержимое с помощью команд ls и cat. (рис. 2.7 , 2.8).

```

func.asm [----] 9 L: [ 1+30 31/ 38] *(471 / 549b) 0078 0x04E
#include 'in_out.asm'
SECTION .data
filename db 'name.txt', 0h
msg db 'Как Вас зовут? ', 0h
strk db 'Меня зовут ', 0h
SECTION .bss
Name resb 255
SECTION .text
global _start
_start:
mov eax, msg
call sprint
mov ecx, Name
mov edx, 255
call sread
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h
mov esi, eax
mov eax, strk
call slen
mov edx, eax
mov ecx, strk
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

```

Рис. 2.7: Написание программы

```

sashubina@dk6n62 ~/work/arch-pc/lab10 $ nasm -f elf func.asm
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o func func.o
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ./func
Как Вас зовут? София Шубина
sashubina@dk6n62 ~/work/arch-pc/lab10 $ ls
func  func.asm  func.o  in_out.asm  lab10-1  lab10-1.asm  lab10-1.o  name.txt  readme-1.txt  readme-2.txt
sashubina@dk6n62 ~/work/arch-pc/lab10 $ cat name.txt
Меня зовут София Шубина
sashubina@dk6n62 ~/work/arch-pc/lab10 $

```

Рис. 2.8: Проверка наличия файла и его содержимого с помощью команд ls, cat

```

#include 'in_out.asm'

SECTION .data

filename db 'name.txt', 0h

msg db 'Как Вас зовут? ', 0h

strk db 'Меня зовут ', 0h

```

```

SECTION .bss
Name resb 255
SECTION .text
global _start
_start:
mov eax,msg
call sprint
mov ecx, Name
mov edx, 255
call sread
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h
mov esi, eax
mov eax,strk
call slen
mov edx, eax
mov ecx, strk
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
```

3 Выводы

Я приобрела навыки написания программ для работы с файлами.

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/Learning-bash-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,
- 11.
12. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
13. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
14. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВ- Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
15. 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.
16. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
17. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер,
18. — 1120 с. — (Классика Computer Science).