

Отчёт по лабораторной работе номер 10

Архитектура компьютера

Волчкова Елизавета Дмитриевна

Содержание

Цель работы

Приобретение навыков написания программ для работы с файлами

Задание

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

Теоретическое введение

С 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] [:новая_группа]` или `chgrp [ключи] < новая_группа >` 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 `г` — 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 `gwx`

(чтение и запись, без исполнения) понимаются как три двоичные цифры 110 или как восьмеричная цифра 6

Выполнение лабораторной работы

Создала каталог для программ лабораторной работы № 10, перешла в него и создала файлы lab10-1.asm, readme-1.txt и readme-2.txt: `mkdir ~/work/arch-pc/lab10 cd ~/work/arch-pc/lab10 touch lab10-1.asm readme-1.txt readme-2.txt`

```
mkdir ~/work/arch-pc/lab10
cd ~/work/arch-pc/lab10|
touch lab10-1.asm readme-1.txt readme-2.txt
```

Затем ввела в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения).

```
ot@liza2006-VirtualBox]:~/work/arch-pc/lab10
/root/work/arch-pc/lab10/lab10-1.asm [-M--] 8 L: [ 1+25 26/ 30] *(1044/1133b) 0059 0x0
SECTION .bss
%include 'in_out.asm'
SECTION .data
filename db 'readme.txt', 0h ; Имя файла
msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
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
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Dele
```

Создала исполняемый файл и проверила его работу.

```

root@liza2006-VirtualBox:~/work/arch-pc/lab10# nasm -f elf32 lab10-1.asm -o obj.o
root@liza2006-VirtualBox:~/work/arch-pc/lab10# ld -m elf_i386 -o lab10 obj.o
root@liza2006-VirtualBox:~/work/arch-pc/lab10# ./lab10
Введите строку для записи в файл: Я люблю Россию!!!

```

Потом с помощью команды `chmod` изменила права доступа к исполняемому файлу `lab10`, запретив его выполнение.

Попыталась выполнить файл - `./lab10`. Объяснение результата:

Код доступа	---	--x	-W-	-WX	r--	r-x	rw-	rwx
Двоичная маска	000	001	010	011	100	101	110	111
Восьмеричная цифра	0	1	2	3	4	5	6	7

С помощью команды `chmod` изменила права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Попыталась выполнить его.

```

root@liza2006-VirtualBox:~/work/arch-pc/lab10# ./lab10
bash: ./lab10: Permission denied

```

```

root@liza2006-VirtualBox:~# mc

root@liza2006-VirtualBox:~/work/arch-pc/lab10# chmod 700 lab10-1.asm

root@liza2006-VirtualBox:~/work/arch-pc/lab10# ls -l lab10-1.asm
-rwx----- 1 root root 1139 Dec 11 00:14 lab10-1.asm

root@liza2006-VirtualBox:~#

```

Объяснение результата: добавила права запуска, теперь файл можно выполнять.

В соответствии с вариантом в таблице 10.4 предоставила права доступа к файлу `readme-1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде.

Проверила правильность выполнения с помощью команды `ls` -

```
root@liza2006-VirtualBox:~/work/arch-pc/lab10# ls -l readme-2.txt
-rwxrwxrwx 1 root root 1139 Dec 10 23:50 readme-2.txt

root@liza2006-VirtualBox:~# mc

root@liza2006-VirtualBox:~/work/arch-pc/lab10# chmod 062 readme-1.txt

root@liza2006-VirtualBox:~/work/arch-pc/lab10# chmod 137 readme-2.txt

root@liza2006-VirtualBox:~/work/arch-pc/lab10# ls -l readme-1.txt
----rw--w- 1 root root 0 Dec 10 23:38 readme-1.txt

root@liza2006-VirtualBox:~/work/arch-pc/lab10# ls -l readme-2.txt
--x-wxrw 1 root root 1139 Dec 10 23:50 readme-2.txt
```

Выполнение самостоятельной работы:

/root/work/arch-pc/lab10/lab10-dz.asm [-M--] 13 L:[1+ 0

section .data

prompt db "Как Вас зовут?", 0

message db "Меня зовут ", 0

filename db "name.txt", 0

buffer resb 100

section .bss

file_handle resd 1

section .text

extern _in_out.asm

global _start

_start:

; Вывод приглашения

call print_string

; Ввод фамилии и имени

call read_string

; Создание файла

mov eax, 5 ; sys_open

mov ebx, filename ; имя файла

mov ecx, 2 ; O_WRONLY | O_CREAT

mov edx, 0644 ; права доступа

int 0x80

mov [file_handle], eax

; Запись в файл

mov eax, 4 ; sys_write

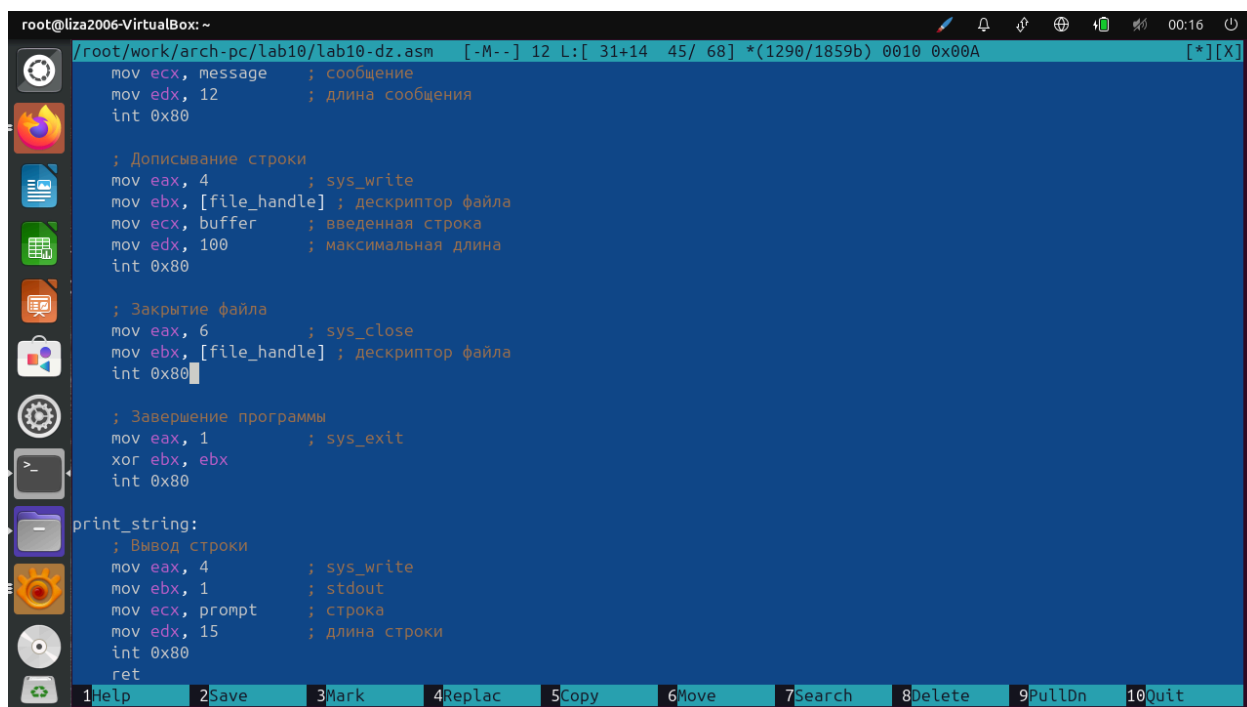
1Help

2Save

3Mark

4Replac

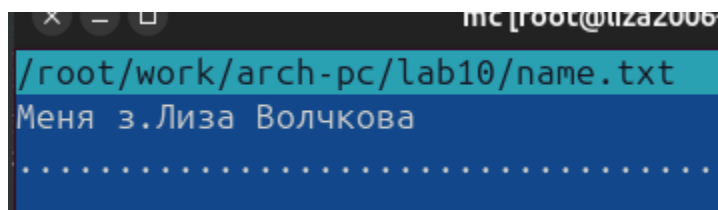
5Copy



```
root@liza2006-VirtualBox: ~  
/root/work/arch-pc/lab10/lab10-dz.asm [-M--] 12 L:[ 31+14 45/ 68] *(1290/1859b) 0010 0x00A [*][X]  
mov ecx, message ; сообщение  
mov edx, 12 ; длина сообщения  
int 0x80  
  
; Дописывание строки  
mov eax, 4 ; sys_write  
mov ebx, [file_handle] ; дескриптор файла  
mov ecx, buffer ; введенная строка  
mov edx, 100 ; максимальная длина  
int 0x80  
  
; Закрытие файла  
mov eax, 6 ; sys_close  
mov ebx, [file_handle] ; дескриптор файла  
int 0x80  
  
; Завершение программы  
mov eax, 1 ; sys_exit  
xor ebx, ebx  
int 0x80  
  
print_string:  
; Вывод строки  
mov eax, 4 ; sys_write  
mov ebx, 1 ; stdout  
mov ecx, prompt ; строка  
mov edx, 15 ; длина строки  
int 0x80  
ret
```

1help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

```
root@liza2006-VirtualBox:~/work/arch-pc/lab10# ./lab10dz  
Как Вас з.Лиза Волчкова
```



```
mc root@liza2006-  
/root/work/arch-pc/lab10/name.txt  
меня з.Лиза Волчкова  
.....
```

Выводы

Целью работы было - приобретение навыков написания программ для работы с файлами, проделав данные задания, я ознакомилась с написанием 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. 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. — БХВ- Петербург, 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,
17. — 1120 с. — (Классика Computer Science).