

Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютеров

Симонова Виктория Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Права доступа к файлам	7
4	Выполнение лабораторной работы	9
4.1	Задание для самостоятельной работы	11
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Создание каталога и файлов	9
4.2	Ввожу код программы в файл	9
4.3	Запускаю исполняемый файл	10
4.4	Изменяю права доступа	10
4.5	Добавляю права доступа	10
4.6	Индивидуальное задание и его проверка	11
4.7	Создаю файл	11
4.8	Задание для самостоятельной работы	12
4.9	Запускаю исполняемый файл и проверяю его работу	12

Список таблиц

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

3.1 Права доступа к файлам

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. Владелец 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 `r` — 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 `rw-` (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная

цифра 6. ## Работа с файлами средствами Nasm

В операционной системе Linux существуют различные методы управления файлами, например, 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 открытии или создании, файлу присваивается уникальный номер (16-битное целое число) – дескриптор файла. Общий алгоритм работы с системными вызовами в Nasm можно представить в следующем виде: 1. Поместить номер системного вызова в регистр EAX; 2. Поместить аргументы системного вызова в регистрах EBX, ECX и EDX; 3. Вызов прерывания (int 80h); 4. Результат обычно возвращается в регистр EAX.

4 Выполнение лабораторной работы

Создаю каталог для программ лабораторной работы № 10, перехожу в него и создаю файлы lab10-1.asm, readme-1.txt и readme-2.txt (рис. [4.1]).

```
[visimonova@fedora ~]$ mkdir ~/work/arch-pc/lab10
[visimonova@fedora ~]$ cd ~/work/arch-pc/lab10
[visimonova@fedora lab10]$ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 4.1: Создание каталога и файлов

Ввожу в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения) (рис. [4.2]).

```
lab10-1.asm      [-M--] 21 L: [ 1+ 0 1/ 40] *(21 /1288b) 0045 0x02D [
;-----
; 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
1Помощь 2Создать 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9Меню
```

Рис. 4.2: Ввожу код программы в файл

Создаю исполняемый файл и проверяю его работу и выполняю инструкции

(рис. [4.3]).

```
[visimonova@fedora lab10]$ nasm -f elf lab10-1.asm
[visimonova@fedora lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[visimonova@fedora lab10]$ ./lab10-1
Введите строку для записи в файл: Hello world!
[visimonova@fedora lab10]$ ls -l
итого 20
-rw-r--r--. 1 visimonova visimonova 3942 ноя  8 07:58 in_out.asm
-rwxr-xr-x. 1 visimonova visimonova 1524 дек 12 15:17 lab10-1
-rw-r--r--. 1 visimonova visimonova 1288 дек 12 15:16 lab10-1.asm
-rw-r--r--. 1 visimonova visimonova 1472 дек 12 15:17 lab10-1.o
-rw-r--r--. 1 visimonova visimonova  13 дек 12 15:18 readme-1.txt
-rw-r--r--. 1 visimonova visimonova   0 дек 11 17:32 readme-2.txt
[visimonova@fedora lab10]$ cat readme-1.txt
Hello world!
```

Рис. 4.3: Запускаю исполняемый файл

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение, затем пытаюсь выполнить файл (рис. [4.4]).

```
[visimonova@fedora lab10]$ chmod u-x lab10-1
[visimonova@fedora lab10]$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис. 4.4: Изменяю права доступа

Исполнить файл не получается, т.к. была введена инструкция `chmod` с аргументами “u” Владелец, “-” Отменить набор прав, “x” Право на исполнение, таким образом я заюрала у файла права на исполнение, поэтому файл и не исполняется.

С помощью команды `chmod` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Пытаюсь выполнить его(рис. [4.5]).

```
[visimonova@fedora lab10]$ chmod u+x lab10-1.asm
[visimonova@fedora lab10]$ ./lab10-1.asm
./lab10-1.asm: строка 1: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 1: `;-----'
```

Рис. 4.5: Добавляю права доступа

Файл не исполняется, т.к. не содержит в себе команд для терминала.

В соответствии с вариантом в таблице 10.4 (мой вариант - 13) предоставляю права доступа к файлу readme-1.txt ,представленные в символьном виде, а для файла readme-2.txt – в двоичном виде. Проверяю правильность выполнения с помощью команды ls -l. (рис. [4.6]).

```
[visimonova@fedora lab10]$ chmod 640 readme-1.txt # -w- --x ---
[visimonova@fedora lab10]$ chmod 640 readme-2.txt # 110 011 001
[visimonova@fedora lab10]$ ls -l
итого 20
-rw-r--r--. 1 visimonova visimonova 3942 ноя  8 07:58 in_out.asm
-rw-r-xr-x. 1 visimonova visimonova 1524 дек 12 15:17 lab10-1
-rwxr--r--. 1 visimonova visimonova 1288 дек 12 15:16 lab10-1.asm
-rw-r--r--. 1 visimonova visimonova 1472 дек 12 15:17 lab10-1.o
-rw-r-----. 1 visimonova visimonova  13 дек 12 15:48 readme-1.txt
-rw-r-----. 1 visimonova visimonova   0 дек 11 17:32 readme-2.txt
```

Рис. 4.6: Индивидуальное задание и его проверка

4.1 Задание для самостоятельной работы

Пишу 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

Создаю файл для выполнения индивидуального задания (рис. [4.7]).

```
[visimonova@fedora lab10]$ touch zadanie1.asm
[visimonova@fedora lab10]$ mc
```

Рис. 4.7: Создаю файл

Ввожу код программы в файл (рис. [4.8]).

```

zadanie1.asm  [----] 21 L:[ 1+ 0 1/ 51] *(21 / 991b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg1 db 'Как Вас зовут?', 0h
filename db 'name.txt', 0h
msg2 db 'Меня зовут ', 0h
SECTION .bss
name resb 255
SECTION .text
global _start
_start:
;-----вывод "как вас зовут"
mov eax,msg1
call sprintf
;-----ввод имени
mov ecx, name
mov edx, 255
call sread
;----- Создание файла
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h

```

Рис. 4.8: Задание для самостоятельной работы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд ls и cat.(рис. [4.9]).

```

zadanie1.asm  [----] 21 L:[ 1+ 0 1/ 51] *(21 / 991b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
msg1 db 'Как Вас зовут?', 0h
filename db 'name.txt', 0h
msg2 db 'Меня зовут ', 0h
SECTION .bss
name resb 255
SECTION .text
global _start
_start:
;-----вывод "как вас зовут"
mov eax,msg1
call sprintf
;-----ввод имени
mov ecx, name
mov edx, 255
call sread
;----- Создание файла
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h

```

Рис. 4.9: Запускаю исполняемый файл и проверяю его работу

Код программы:

```

%include 'in_out.asm'

SECTION .data
msg1 db 'Как Вас зовут?', 0h
filename db 'name.txt', 0h
msg2 db 'Меня зовут ', 0h

SECTION .bss
name resb 255

SECTION .text
global _start

_start:
;-----ВЫВОД "как вас зовут"
mov eax,msg1
call sprintLF
;-----ВВОД ИМЕНИ
mov ecx, name
mov edx, 255
call sread
;----- Создание файла
mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h
;-----открытие файла
mov ecx, 2
mov ebx, filename
mov eax, 5
int 80h
; --- Запись дескриптора файла в `esi`
mov esi, eax

```

```
; --- Расчет длины введенной строки
mov eax, msg2
call slen
; --- Записываем в файл
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
; --- Закрываем файл (`sys_close`)
mov ebx, esi
mov eax, 6
int 80h
call quit
```

5 Выводы

Приобрела навыки написания программ для работы с файлами, разобралась в правах доступа к файлам.

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,
- 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. — БХВПетербург, 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).