

Отчёта по лабораторной работе №10

Простейший вариант

Сахно Алёна Юрьевна

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	Создание каталога	8
2.2	Вывод файлов	9
2.3	Название рисунка	10
2.4	Изменение файлов	10
2.5	Права доступа	11

Список таблиц

1 Цель работы

Приобретение навыков написания программ для работы с файлами. # Зада-
ние 1. Теоретическая часть 2. Выполнения лабораторной работы 3. Выполнение
самостоятельной работы

2 Теоретическое введение

10.2. Теоретическое введение

10.2.1. 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. 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 `х` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `гw` (чтение и запись, без исполнения) понимаются как три двоичные цифры 110

или как восьмеричная цифра 6.

Полная строка прав доступа в символьном представлении имеет вид: Так, например, права `gwx r-x -x` выглядят как двоичное число `111 101 001`, или восьмеричное `751`. Свойства (атрибуты) файлов и каталогов можно вывести на терминал с помощью команды `ls` с ключом `-l`. Так например, чтобы узнать права доступа к файлу `README` можно узнать с помощью следующей команды: `$ls -l /home/debugger/README -rwxr-xr- 1 debugger users 0 Feb 14 19:08 /home/debugger/README` В первой колонке показаны текущие права доступа, далее указан владелец файла и группа: Тип файла определяется первой позицией, это может быть: каталог — `d`, обычный файл — дефис (`-`) или символьная ссылка на другой файл — `l`. Следующие 3 набора по 3 символа определяют конкретные права для конкретных групп: `r` — разрешено чтение файла, `w` — разрешена запись в файл; `x` — разрешено исполнение файла и дефис (`-`) — право не дано. Для изменения прав доступа служит команда `chmod`, которая понимает как символьное, так и числовое указание прав. Для того чтобы назначить файлу `/home/debugger/README` права `rw-r`, то есть разрешить владельцу чтение и запись, группе только чтение, остальным пользователям — ничего:

```
$chmod 640 README # 110 100 000 == 640 == rw-r-- $ls -l README -rw-r 1
debugger users 0 Feb 14 19:08 /home/debugger/README
```

В символьном представлении есть возможность явно указывать какой группе какие права необходимо добавить, отнять или присвоить. Например, чтобы добавить право на исполнение файла `README` группе и всем остальным:

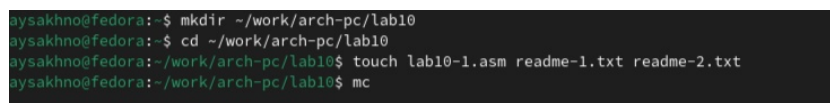
```
$chmod go+x README $ls -l README -rw-r-x-x 1 debugger users 0 Feb 14 19:08
/home/debugger/README
```

Формат символьного режима: `chmod` Возможные значения аргументов команды представлены в таблице 10.2 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 возвращает-

ся в регистр EAX. # Выполнение лабораторной работы 1. Создайте каталог для программ лабораторной работы № 10, перейдите в него и создайте 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.1 2.1).

A screenshot of a terminal window showing the following commands and their output:

```
aysakhno@fedora:~$ mkdir ~/work/arch-pc/lab10
aysakhno@fedora:~$ cd ~/work/arch-pc/lab10
aysakhno@fedora:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
aysakhno@fedora:~/work/arch-pc/lab10$ mc
```

Рис. 2.1: Создание каталога

2. Введите в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения). Создайте исполняемый файл и проверьте его работу.

(рис.2 2.2).



```
GNU nano 7.2 /home/aysakhno/work/arch-pc/lab10/lab10-1.asm
-----
Запись в файл строки введенной на запрос
-----
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
```

Рис. 2.2: Вывод файлов

3. С помощью команды `chmod` измените права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. Попробуйте выполнить файл. Объясните результат.

(рис. 2.3).

```
aysakhno@fedora:~/work/arch-pc/lab10
aysakhno@fedora:~/work/arch-pc/lab10$ mc
aysakhno@fedora:~/work/arch-pc/lab10$ cat readme-1.txt
aysakhno@fedora:~/work/arch-pc/lab10$ cat readme-1.txt
aysakhno@fedora:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
aysakhno@fedora:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
aysakhno@fedora:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: sakhno
aysakhno@fedora:~/work/arch-pc/lab10$ ls -l
итого 40
-rw-r--r--. 1 aysakhno aysakhno 3942 ноя 14 13:13 in_out.asm
-rwxr-xr-x. 1 aysakhno aysakhno 9736 дек 14 13:15 lab10-1
-rw-r--r--. 1 aysakhno aysakhno 1287 дек 14 12:59 lab10-1.asm
-rw-r--r--. 1 aysakhno aysakhno 13713 дек 14 13:15 lab10-1.lst
-rw-r--r--. 1 aysakhno aysakhno 2512 дек 14 13:15 lab10-1.o
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-1.txt
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-2.txt
aysakhno@fedora:~/work/arch-pc/lab10$ cat readme-1.txt
aysakhno@fedora:~/work/arch-pc/lab10$
```

Рис. 2.3: Название рисунка

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.

(рис.4 2.4).

```
aysakhno@fedora:~/work/arch-pc/lab10
-rwxr-xr-x. 1 aysakhno aysakhno 9736 дек 14 13:15 lab10-1
-rw-r--r--. 1 aysakhno aysakhno 1287 дек 14 12:59 lab10-1.asm
-rw-r--r--. 1 aysakhno aysakhno 13713 дек 14 13:15 lab10-1.lst
-rw-r--r--. 1 aysakhno aysakhno 2512 дек 14 13:15 lab10-1.o
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-1.txt
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-2.txt
aysakhno@fedora:~/work/arch-pc/lab10$ chmod u-x lab10-1
aysakhno@fedora:~/work/arch-pc/lab10$ ls -l
итого 40
-rw-r--r--. 1 aysakhno aysakhno 3942 ноя 14 13:13 in_out.asm
-rwxr-xr-x. 1 aysakhno aysakhno 9736 дек 14 13:15 lab10-1
-rw-r--r--. 1 aysakhno aysakhno 1287 дек 14 12:59 lab10-1.asm
-rw-r--r--. 1 aysakhno aysakhno 13713 дек 14 13:15 lab10-1.lst
-rw-r--r--. 1 aysakhno aysakhno 2512 дек 14 13:15 lab10-1.o
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-1.txt
-rw-r--r--. 1 aysakhno aysakhno 0 дек 14 12:57 readme-2.txt
aysakhno@fedora:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
aysakhno@fedora:~/work/arch-pc/lab10$
```

Рис. 2.4: Изменение файлов

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. Проверить правильность выполнения с помощью команды `ls -l`.

(рис.5 2.5).

```
aysakhno@fedora:~/work/arch-pc/lab10$ chmod u=rx,g=rwx,o=rwx readme-1.txt  
aysakhno@fedora:~/work/arch-pc/lab10$ chmod 676 readme-2.txt  
aysakhno@fedora:~/work/arch-pc/lab10$
```

Рис. 2.5: Права доступа

У меня вариант 14 r-x rwx rwx 110 111 110

3 Выводы

Я приобрела навык написания программ для работы с файлами.

Список литературы