

Лабораторная работа №10

Работа с файлами средствами Nasm

Карпова Есения Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	15
	Список литературы	16

Список иллюстраций

4.1	Программа записи в файл сообщения	10
4.2	Запуск исполняемого файла	11
4.3	Запрет на выполнение файла	11
4.4	Добавление права на исполнение	11
4.5	Предоставление прав доступа	12
4.6	Текст программы	13
4.7	Запуск исполняемого файла	14

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

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 Обработка файлов в операционной системе Linux осуществляется за счет использования определенных системных вызовов: - 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 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. - 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`. - Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре `EBX`.

4 Выполнение лабораторной работы

1. Написание программ для работы с файлами Создаю каталог для программ лабораторной работы № 10 и файлы lab10-1.asm, readme-1.txt и readme-2.txt и ввожу в файл lab10-1.asm текст программы, записанный в листинге 10.1 (рис. 4.2).

```

;-----
; 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
mov ecx, 2
mov ebx, filename
mov eax, 5
int 80h

; --- Запись дескриптора файла в 'esi'

```

Рис. 4.1: Программа записи в файл сообщения

Создаю исполняемый файл и проверяю его работу. Убеждаюсь, что файл работает корректно. (рис. ??).

```
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ nasm -f elf lab10-1.asm
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-1 lab10-1.o
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ./lab10-1
Введите строку для записи в файл: I love you
```

Рис. 4.2: Запуск исполняемого файла

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение, и пытаюсь выполнить файл (рис. 4.3).

```
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ chmod u-x lab10-1
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ./lab10-1
bash: ./lab10-1: Отказано в доступе
```

Рис. 4.3: Запрет на выполнение файла

Программа пишет: “Отказано в доступе”. Файл не выполнился, так как в команде указана отмена прав (“-”) на исполнение (“x”) владельцу (“u”).

С помощью команды `chmod` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение (рис. 4.2).

```
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ chmod u+x lab10-1.asm
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ./lab10-1.asm
./lab10-1.asm: строка 1: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 1: `;-----'
```

Рис. 4.4: Добавление права на исполнение

Текстовый файл начинает исполнение, но не исполняется, так как не содержит в себе команд для терминала.

В соответствии со своим вариантом (9) представляю права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme2.txt` - в двоичном, затем проверяю правильность выполнения (рис. 4.5).

```

eakarpova@dk5n58 ~/work/arch-pc/lab10 $ chmod 640 readme-1.txt # --x -w- -w-
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ chmod 640 readme-2.txt # 001 011 101
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ls -l
итого 17
-rw-r--r-- 1 eakarpova studsci 3942 ноя 10 12:42 in_out.asm
-rw-r-xr-x 1 eakarpova studsci 9164 дек 12 10:58 lab10-1
-rwxr--r-- 1 eakarpova studsci 1216 дек 12 10:54 lab10-1.asm
-rw-r--r-- 1 eakarpova studsci 1472 дек 12 10:58 lab10-1.o
-rw-r----- 1 eakarpova studsci  0 дек 12 10:51 readme-1.txt
-rw-r----- 1 eakarpova studsci  0 дек 12 10:51 readme-2.txt

```

Рис. 4.5: Предоставление прав доступа

2. Задание для самостоятельной работы

Пишу код программы, выводящей приглашение “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя, и создающую файл, в который записывается сообщение “Меня зовут”(рис. 4.6).

```

%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:
mov eax, msg1
call sprintf

mov ecx, name
mov edx, 255
call sread

mov ecx, 0777o
mov ebx, filename
mov eax, 8
int 80h

mov ecx, 2
mov ebx, filename
mov eax, 5
int 80h

mov esi, eax

```

Рис. 4.6: Текст программы

Создаю исполняемый файл и проверяю его работу, после этого проверяю наличие файла и его содержимое (рис. 4.7).

```
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ nasm -f elf lab10-2.asm
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ld -m elf_i386 -o lab10-2 lab10-2.o
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ls
in_out.asm  lab10-1.asm  lab10-2      lab10-2.o    readme-2.txt
lab10-1     lab10-1.o    lab10-2.asm  readme-1.txt
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ./lab10-2
Как Вас зовут?
Карпова Есения
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ ls
in_out.asm  lab10-1.asm  lab10-2      lab10-2.o    readme-1.txt
lab10-1     lab10-1.o    lab10-2.asm  name.txt     readme-2.txt
eakarpova@dk5n58 ~/work/arch-pc/lab10 $ cat name.txt
Меня зовут Карпова Есения
```

Рис. 4.7: Запуск исполняемого файла

5 Выводы

В ходе выполнения лабораторной работы я приобрела навыки написания программ для работы с файлами.

Список литературы

Демидова А.В. - Лабораторная работа №10. Работа с файлами средствами Nasm