

Отчет по лабораторной работе №10

Дисциплина: архитектура компьютера

Гомазкова Алина

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Написание программ для работы с файлами	9
4.2	Задание для самостоятельной работы	11
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Рис. 1 Создание файлов для лабораторной работы	9
4.2	Рис. 2 Ввод текста из листинга 10.1	10
4.3	Рис. 3 Запуск исполняемого файла	10
4.4	Рис. 4 Запрет на выполнение файла	10
4.5	Рис. 5 Добавление прав на исполнение	11
4.6	Рис. 6 Предоставление прав доступа в символьном и двоичном виде	11
4.7	Рис. 7 Написание текста программы	12
4.8	Рис. 8 Загрузка исполняемого файла и проверка его работы	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

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.

Для изменения прав доступа служит команда `chmod`, которая понимает как символьное, так и числовое указание прав.

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.

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 открытия существующего файла служит системный вызов `sys_open`, который использует следующие аргументы: права доступа к файлу в регистре `EDX`, режим доступа к файлу в регистр `ECX`, имя файла в `EBX` и номер системного вызова `sys_open` (5) 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`. Системный вызов возвращает фактическое количество

записанных байтов в регистр EAX. В случае ошибки, код ошибки также будет находиться в регистре EAX. Прежде чем записывать в файл, его необходимо создать или открыть, что позволит получить дескриптор файла.

Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре EDI, 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 EDI, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова `sys_lseek` (19) в EAX. Значение смещения можно задавать в байтах.

Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре EBX.

4 Выполнение лабораторной работы

4.1 Написание программ для работы с файлами

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

```
[alinagomazkova@10 ~]$ mkdir -p ~/work/arch-pc/lab10  
[alinagomazkova@10 ~]$ cd ~/work/arch-pc/lab10  
[alinagomazkova@10 lab10]$ touch lab10-1.asm readme-1.txt readme-2.txt
```

Рис. 4.1: Рис. 1 Создание файлов для лабораторной работы

Ввожу в файл lab10-1.asm текст программы, записывающей в файл сообщения, из листинга 10.1 (рис. 2)

```

mc [alinagomazkova@10.0.2.15]:~/work/arch-pc/lab10
lab10-1.asm  [-M--]  0 L:[ 1+ 0  1/ 40] *(0  /1286b) 0059 0x03B
; 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')

```

Рис. 4.2: Рис. 2 Ввод текста из листинга 10.1

Создаю исполняемый файл и проверяю его работу (рис. 3)

```

[alinagomazkova@10 lab10]$ nasm -f elf lab10-1.asm
[alinagomazkova@10 lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[alinagomazkova@10 lab10]$ ./lab10-1
Введите строку для записи в файл: Hello world!
[alinagomazkova@10 lab10]$ cat readme-1.txt
Hello world!

```

Рис. 4.3: Рис. 3 Запуск исполняемого файла

Далее с помощью команды `chmod u-x` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение и пытаюсь выполнить файл (рис. 4)

```

[alinagomazkova@10 lab10]$ chmod u-x lab10-1
[alinagomazkova@10 lab10]$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
[alinagomazkova@10 lab10]$

```

Рис. 4.4: Рис. 4 Запрет на выполнение файла

Файл не выполняется, т.к в команде я указала “u” - владелец (себя), “-” - отменить набор прав, “x” - право на исполнение.

С помощью команды `chmod u+x lab10-1.asm` изменяю права доступа к файлу `lab10-1.asm` исходным текстом программы, добавив права на исполнение, и пытаюсь выполнить его (рис. 5)

```
[alinagomazkova@i0 lab10]$ chmod u+x lab10-1.asm
[alinagomazkova@i0 lab10]$ ./lab10-1.asm
./lab10-1.asm: строка 1: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 1: `;-----'
[alinagomazkova@i0 lab10]$
```

Рис. 4.5: Рис. 5 Добавление прав на исполнение

Текстовый файл начинает исполнение, но не исполняется, т.к не содержит в себе команд для терминала.

В соответствии со своим вариантом (9) в таблице 10.4 предоставляю права доступа к файлу `readme1.txt` представленные в символьном виде, а для файла `readme-2.txt` – в двоичном виде:

`-x -w- -w- , 001 011 101`

И проверяю правильность выполнения с помощью команды `ls -l` (рис. 6)

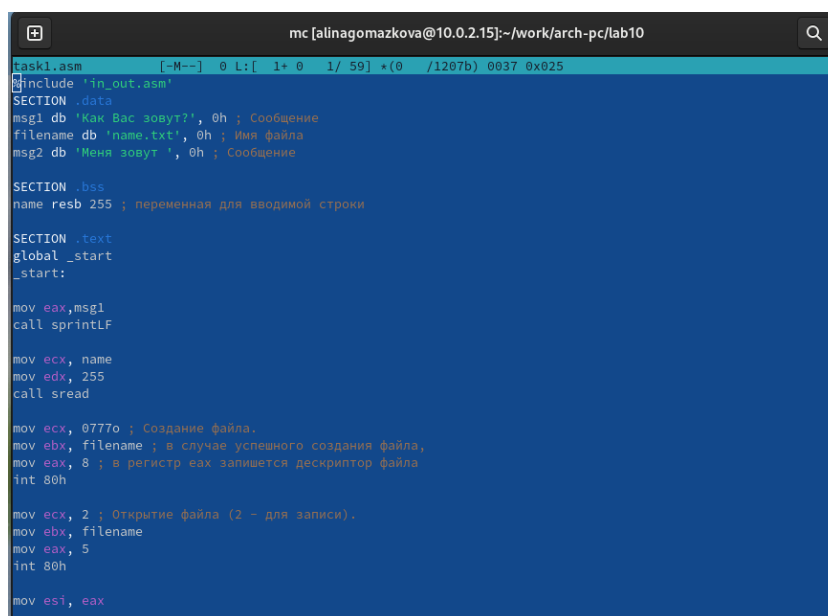
```
[alinagomazkova@i0 lab10]$ chmod 640 readme-1.txt # --x -w- -w-
[alinagomazkova@i0 lab10]$ chmod 640 readme-2.txt # 001 011 101
[alinagomazkova@i0 lab10]$ ls -l
итого 32
-rw-r--r--. 1 alinagomazkova alinagomazkova 3942 ноя 15 12:56 in_out.asm
-rw-r-xr-x. 1 alinagomazkova alinagomazkova 9164 дек 16 16:50 lab10-1
-rwxr--r--. 1 alinagomazkova alinagomazkova 1288 дек 16 16:50 lab10-1.asm
-rw-r--r--. 1 alinagomazkova alinagomazkova 1472 дек 16 16:50 lab10-1.o
-rw-r-----. 1 alinagomazkova alinagomazkova 13 дек 16 16:51 readme-1.txt
-rw-r-----. 1 alinagomazkova alinagomazkova 0 дек 16 16:49 readme-2.txt
-rw-r--r--. 1 alinagomazkova alinagomazkova 1136 дек 16 17:02 task1.asm
[alinagomazkova@i0 lab10]$
```

Рис. 4.6: Рис. 6 Предоставление прав доступа в символьном и двоичном виде

4.2 Задание для самостоятельной работы

Пишу код программы, выводящей приглашения “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя и создающую файл, в который записывается

сообщение “Меня зовут”ФИ”” (рис. 7)



```
task1.asm [-M--] 0 L:[ 1+ 0 1/ 59] *(0 /1207b) 0037 0x025
#include 'in_out.asm'
SECTION .data
msg1 db 'Как Вас зовут?', 0h ; Сообщение
filename db 'name.txt', 0h ; Имя файла
msg2 db 'Меня зовут ', 0h ; Сообщение

SECTION .bss
name resb 255 ; переменная для вводимой строки

SECTION .text
global _start
_start:

mov eax, msg1
call sprintf

mov ecx, name
mov edx, 255
call sread

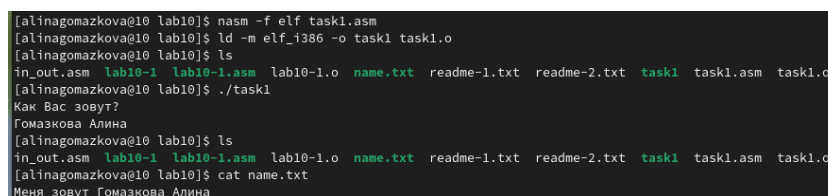
mov ecx, 0777o ; Создание файла.
mov ebx, filename ; в случае успешного создания файла,
mov eax, 8 ; в регистр eax запишется дескриптор файла
int 80h

mov ecx, 2 ; Открытие файла (2 - для записи).
mov ebx, filename
mov eax, 5
int 80h

mov esi, eax
```

Рис. 4.7: Рис. 7 Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд ls и cat (рис. 8)



```
[alinagomazkova@i0 lab10]$ nasm -f elf task1.asm
[alinagomazkova@i0 lab10]$ ld -m elf_i386 -o task1 task1.o
[alinagomazkova@i0 lab10]$ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.o name.txt readme-1.txt readme-2.txt task1 task1.asm task1.o
[alinagomazkova@i0 lab10]$ ./task1
Как Вас зовут?
Гомазкова Алина
[alinagomazkova@i0 lab10]$ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.o name.txt readme-1.txt readme-2.txt task1 task1.asm task1.o
[alinagomazkova@i0 lab10]$ cat name.txt
Меня зовут Гомазкова Алина
```

Рис. 4.8: Рис. 8 Загрузка исполняемого файла и проверка его работы

Программа работает.

5 Выводы

Благодаря данной лабораторной работе я приобрела навыки написания программ для работы с файлами.

Список литературы

1.Архитектура ЭВМ