

Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
4.1	Написание программ для работы с файлами	8
4.2	Выполнение заданий для самостоятельной работы	11
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Создание файлов для лабораторной работы	8
4.2	Ввод текста программы	9
4.3	Запуск исполняемого файла	9
4.4	Запрет на выполнение файла	10
4.5	Добавление прав на исполнение	10
4.6	Предоставление прав доступа в символьном и двоичном виде . .	10
4.7	Написание текста программы	12
4.8	Запуск исполняемого файла	13

1 Цель работы

Приобрести навыки написания программ для работы с файлами.

2 Задание

1. Написание программ для работы с файлами
2. Выполнение заданий 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 `syscreat`, который использует следующие аргументы: права доступа к файлу в регистре `ECX`, имя файла в `EBX` и номер системного вызова `sys_creat` в `EAX`. Для открытия существующего файла служит системный вызов `sysopen`, 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 `syswrite`, который использует следующие аргументы: количество байтов для записи в регистре `EDX`, строку содержимого для записи в `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 `sysread`, 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 `syslseek`, 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`. 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

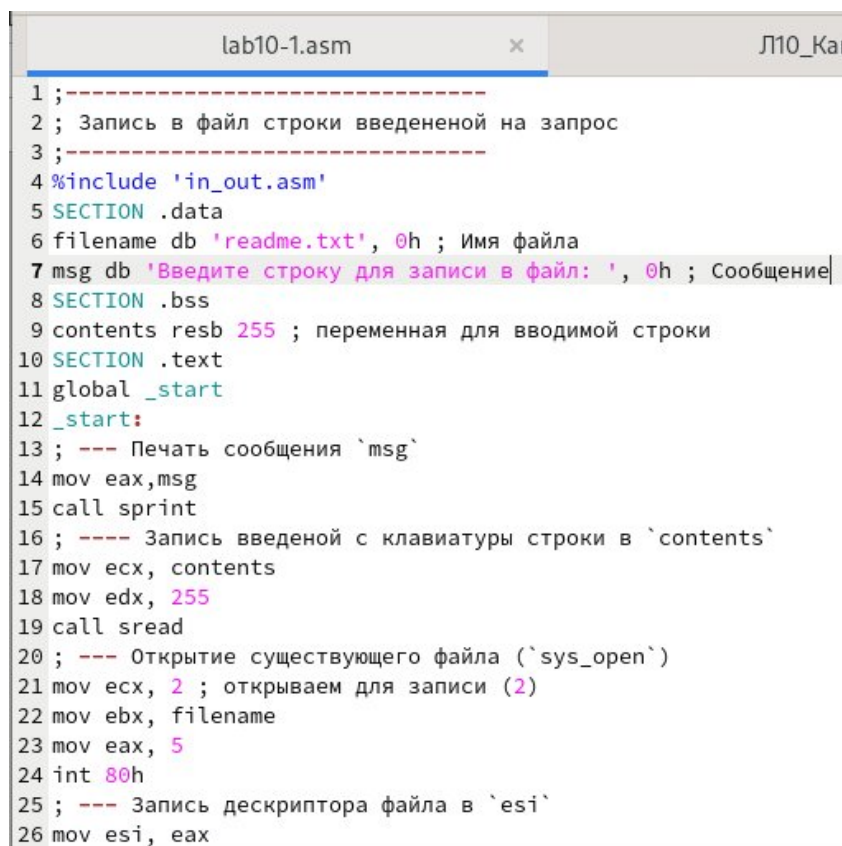
4.1 Написание программ для работы с файлами

Создаю каталог для данной лабораторной работы и уже в нём создаю файлы lab10-1.asm, readme-1.txt, readme-2.txt

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

Рис. 4.1: Создание файлов для лабораторной работы

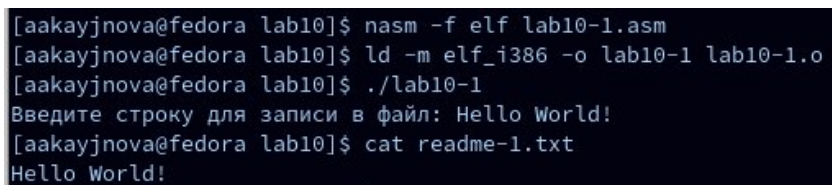
Ввожу в файл lab10-1.asm код программы из листинга 10.1



```
lab10-1.asm x Л10_Ка
1 ;-----
2 ; Запись в файл строки введенной на запрос
3 ;-----
4 %include 'in_out.asm'
5 SECTION .data
6 filename db 'readme.txt', 0h ; Имя файла
7 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
8 SECTION .bss
9 contents resb 255 ; переменная для вводимой строки
10 SECTION .text
11 global _start
12 _start:
13 ; --- Печать сообщения `msg`
14 mov eax,msg
15 call sprint
16 ; ---- Запись введенной с клавиатуры строки в `contents`
17 mov ecx, contents
18 mov edx, 255
19 call sread
20 ; --- Открытие существующего файла (`sys_open`)
21 mov ecx, 2 ; открываем для записи (2)
22 mov ebx, filename
23 mov eax, 5
24 int 80h
25 ; --- Запись дескриптора файла в `esi`
26 mov esi, eax
```

Рис. 4.2: Ввод текста программы

Создаю исполняемый файл и проверяю его работу



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

Рис. 4.3: Запуск исполняемого файла

Изменяю права доступа к исполняемому файлу lab10-1, запретив его выполнение и проверяю правильность запрета

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

Рис. 4.4: Запрет на выполнение файла

Файл не выполняется, так как в команде указали “u” - то есть владельца(себя) и “-” - отменить набор прав, “x” - право на исполнение.

Изменяю права доступа к данному файлу с исходным текстом программы, добавив права на исполнение и проверяю правильность добавления прав

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

Рис. 4.5: Добавление прав на исполнение

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

В соответствии со своим вариантом (5) в таблице 10.4 предоставляю права доступа к файлу readme-1.txt, представленные в символьном виде (-x -w- r-x), а для файла readme-2.txt - в двоичном виде (001 101 010) и проверяю правильность выполнения с помощью команды ls -l

```
[aakayjnova@fedora lab10]$ chmod 640 readme-1.txt # --x-w-r-x
[aakayjnova@fedora lab10]$ chmod 640 readme-2.txt # 001 101 010
[aakayjnova@fedora lab10]$ ls -l
итого 28
-rw-r--r--. 1 aakayjnova aakayjnova 3942 окт 21 22:13 in_out.asm
-rw-r-xr-x. 1 aakayjnova aakayjnova 9164 дек 16 15:38 lab10-1
-rwxr--r--. 1 aakayjnova aakayjnova 1290 дек 16 15:38 lab10-1.asm
-rw-r--r--. 1 aakayjnova aakayjnova 1472 дек 16 15:38 lab10-1.o
-rw-r-----. 1 aakayjnova aakayjnova 13 дек 16 15:39 readme-1.txt
-rw-r-----. 1 aakayjnova aakayjnova 0 дек 16 15:17 readme-2.txt
```

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

4.2 Выполнение заданий для самостоятельной работы

Пишу код программы, выводящей предложение “Как Вас зовут?”, считывающей с клавиатуры ФИ, создающую файл, в который записывается сообщение “Меня зовут ФИ”

```

1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Как Вас зовут?',0h
4 filename db 'name.txt',0h
5 msg2 db 'Меня зовут ',0h
6 SECTION .bss
7 name resb 255
8 SECTION .text
9 global _start
10 _start:
11 mov eax,msg1
12 call sprintLF
13 mov ecx,name
14 mov edx,255
15 call sread
16 mov ecx,0777o
17 mov ebx,filename
18 mov eax,8
19 int 80h
20 mov ecx,2
21 mov ebx,filename
22 mov eax,5
23 int 80h
24 mov esi,eax
25 mov eax,msg2
26 call slen
27 mov edx,eax
28 mov ecx,msg2
29 mov ebx,esi
30 mov eax,4
31 int 80h
32 mov eax,name
33 call slen
34 mov edx,eax
35 mov ecx,name
36 mov ebx,esi
37 mov eax,4
38 int 80h
39 mov ebx,esi
40 mov eax,6
41 int 80h
42 call quit

```

Рис. 4.7: Написание текста программы

Создаю исполняемый файл и проверяю его работу. Проверяю наличие файла и его содержимое с помощью команд `ls` и `cat`

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

Рис. 4.8: Запуск исполняемого файла

Программа работает корректно.

5 Выводы

В ходе данной лабораторной работы мы приобрели навыки написания программ для работы с файлами.

Список литературы

- [illegible]