

Отчёта по лабораторной работе №10

Работа с файлами средствами Nasm.

Спелов Андрей Николаевич

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Задание для самостоятельной работы	9
3	Выводы	12

Список иллюстраций

2.1	Создаем каталог с помощью команды <code>mkdir</code> и файлы с помощью команды <code>touch</code>	5
2.2	Заполняем файл	6
2.3	Запускаем файл и проверяем его работу	6
2.4	Используем команду <code>chmod</code> для установки нужных прав, после этого пытаемся запустить файл	7
2.5	Используем команду <code>chmod</code> для установки нужных прав, после этого пытаемся запустить файл	8
2.6	Используем команду <code>chmod</code> для установки нужных прав, после этого проверяем правильность выполнения командой <code>ls -l</code>	9
2.7	Создаем файл командой <code>touch</code>	9
2.8	Пишем программу в <code>midnight commander</code>	10
2.9	Проверяем работу программы	11

1 Цель работы

Приобрести навыки написания программ для работы с файлам, научиться управлять доступом к файлам.

2 Выполнение лабораторной работы

Создаем каталог для программ ЛБ10, и в нем создаем файлы (рис. 2.1).

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

Рис. 2.1: Создаем каталог с помощью команды `mkdir` и файлы с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 10.1 (рис. 2.2).

```

lab10-1.asm      [-M--] 11 L:[ 1+17 18/ 30] *(360 / 554b) 0010 0x00A
#include 'in_out.asm'
SECTION .data
filename db 'readme.txt', 0h
msg db 'Введите строку для записи в файл: ', 0h
SECTION .bss
contents resb 255
SECTION .text
global _start
_start:
    mov eax,msg
    call sprint
    mov ecx, contents
    mov edx, 255
    call sread
    mov ecx, 2
    mov ebx, filename
    mov eax, 5
    int 80h
    mov esi, eax
    mov eax, contents
    call slen
    mov edx, eax
    mov ecx, contents
    mov ebx, esi
    mov eax, 4
    int 80h
    mov ebx, esi
    mov eax, 6
    int 80h
    call quit

```

Рис. 2.2: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 2.3).

```

[spelovandrei@fedora lab10]$ nasm -f elf -g -l main.lst main.asm
nasm: fatal: unable to open input file 'main.asm' No such file or directory
[spelovandrei@fedora lab10]$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
[spelovandrei@fedora lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[spelovandrei@fedora lab10]$ ./lab10-1
Введите строку для записи в файл: Hello world!
[spelovandrei@fedora lab10]$

```

Рис. 2.3: Запускаем файл и проверяем его работу

Изменяем права доступа к файлу, запретив его выполнение. Пробуем запустить файл (рис. 2.4).

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

Рис. 2.4: Используем команду `chmod` для установки нужных прав, после этого пытаемся запустить файл

Выдало: отказано в доступе. Значит мы поставили правильный запрет на выполнение.

Изменяем права доступа к файлу с исходным текстом программы, добавив права на исполнение. Пробуем запустить файл (рис. 2.5).

```

[spelovandrei@fedora lab10]$ chmod +x lab10-1.asm
[spelovandrei@fedora lab10]$ ./lab10-1.asm
./lab10-1.asm: строка 1: fg: нет управления заданиями
./lab10-1.asm: строка 2: SECTION: команда не найдена
./lab10-1.asm: строка 3: filename: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 7: SECTION: команда не найдена
./lab10-1.asm: строка 8: global: команда не найдена
./lab10-1.asm: строка 9: _start:: команда не найдена
./lab10-1.asm: строка 10: mov: команда не найдена
./lab10-1.asm: строка 11: call: команда не найдена
./lab10-1.asm: строка 12: mov: команда не найдена
./lab10-1.asm: строка 13: mov: команда не найдена
./lab10-1.asm: строка 14: call: команда не найдена
./lab10-1.asm: строка 15: mov: команда не найдена
./lab10-1.asm: строка 16: mov: команда не найдена
./lab10-1.asm: строка 17: mov: команда не найдена
./lab10-1.asm: строка 18: int: команда не найдена
./lab10-1.asm: строка 19: mov: команда не найдена
./lab10-1.asm: строка 20: mov: команда не найдена
./lab10-1.asm: строка 21: call: команда не найдена
./lab10-1.asm: строка 22: mov: команда не найдена
./lab10-1.asm: строка 23: mov: команда не найдена
./lab10-1.asm: строка 24: mov: команда не найдена
./lab10-1.asm: строка 25: mov: команда не найдена
./lab10-1.asm: строка 26: int: команда не найдена
./lab10-1.asm: строка 27: mov: команда не найдена
./lab10-1.asm: строка 28: mov: команда не найдена
./lab10-1.asm: строка 29: int: команда не найдена
./lab10-1.asm: строка 30: call: команда не найдена
[spelovandrei@fedora lab10]$

```

Рис. 2.5: Используем команду `chmod` для установки нужных прав, после этого пытаемся запустить файл

`lab10-1.asm` является файлом с исходным кодом программы на языке ассемблера, искусственно добавление права на исполнение не даст ожидаемого результата. Такие файлы нужно компилировать или ассемблировать в машинный код, а затем выполнять.

ВАРИАНТ 20

Предоставляем права доступа к 2ум файлам, согласно варианту 20 в символ-

ном и двоичном виде, затем проверяем работу команд. (рис. 2.6).

```
[spelovandrei@fedora lab10]$ chmod u=g,rw,o=w readme-1.txt
[spelovandrei@fedora lab10]$ chmod 137 readme-2.txt
[spelovandrei@fedora lab10]$ ls -l
итого 40
-rw-r--r--. 1 spelovandrei spelovandrei 3942 окт 26 20:31 in_out.asm
-rwxr-xr-x. 1 spelovandrei spelovandrei 9740 дек 5 01:30 lab10-1
-rwxr-xr-x. 1 spelovandrei spelovandrei 554 дек 5 01:13 lab10-1.asm
-rw-r--r--. 1 spelovandrei spelovandrei 12581 дек 5 01:29 lab10-1.lst
-rw-r--r--. 1 spelovandrei spelovandrei 2528 дек 5 01:29 lab10-1.o
----rw--w-. 1 spelovandrei spelovandrei 0 дек 5 01:04 readme-1.txt
---x-wrwx. 1 spelovandrei spelovandrei 0 дек 5 01:04 readme-2.txt
[spelovandrei@fedora lab10]$
```

Рис. 2.6: Используем команду `chmod` для установки нужных прав, после этого проверяем правильность выполнения командой `ls -l`

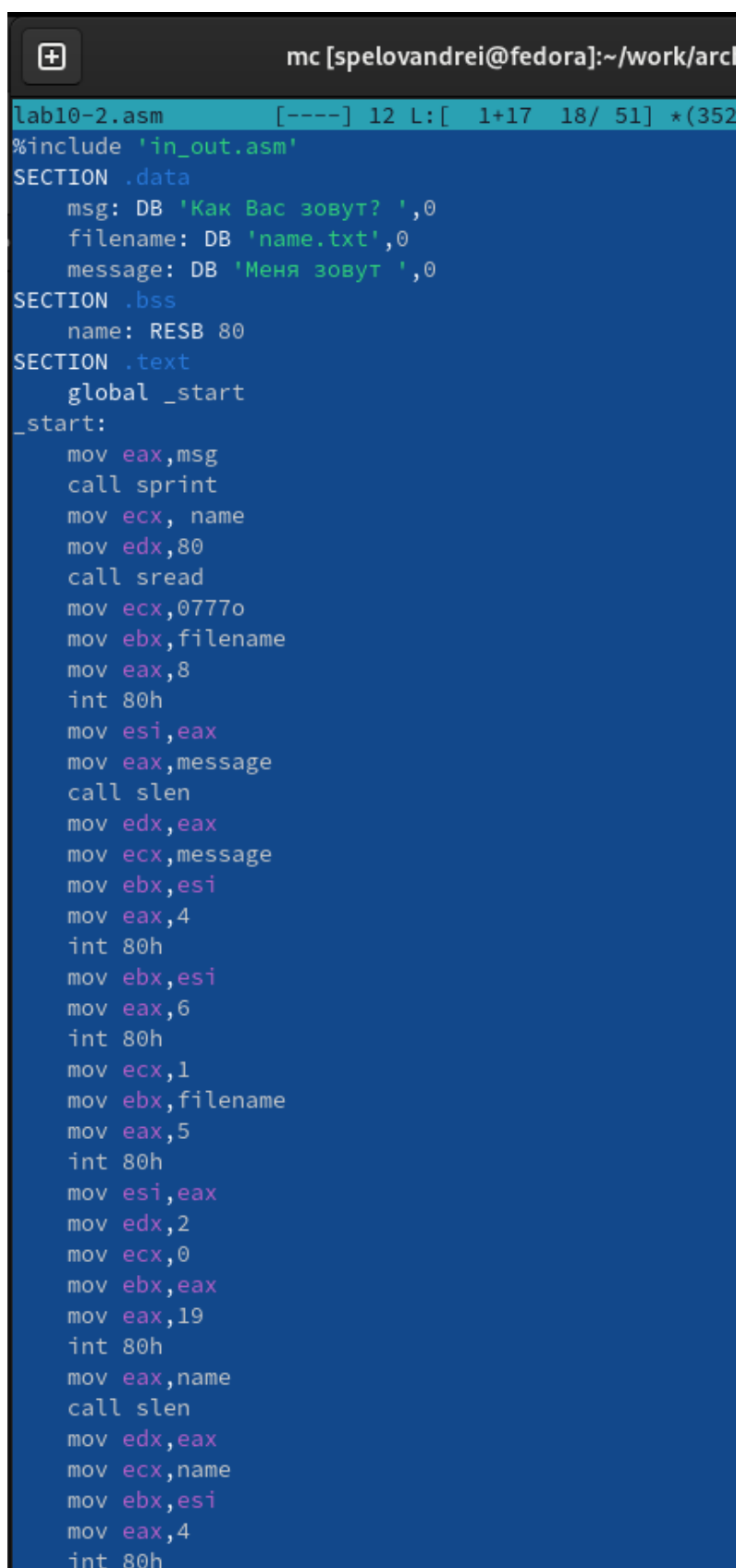
2.1 Задание для самостоятельной работы

Создаем новый файл (рис. 2.7).

```
[spelovandrei@fedora lab10]$ touch lab10-2.asm
[spelovandrei@fedora lab10]$
```

Рис. 2.7: Создаем файл командой `touch`

Пишем программу, которая выполнит представленный список действий (рис. 2.8).



```
mc [spelovandrei@fedora]:~/work/arc
lab10-2.asm [----] 12 L:[ 1+17 18/ 51] *(352
#include 'in_out.asm'
SECTION .data
    msg: DB 'Как Вас зовут? ',0
    filename: DB 'name.txt',0
    message: DB 'Меня зовут ',0
SECTION .bss
    name: RESB 80
SECTION .text
    global _start
_start:
    mov eax,msg
    call sprint
    mov ecx, name
    mov edx,80
    call sread
    mov ecx,0777o
    mov ebx,filename
    mov eax,8
    int 80h
    mov esi,eax
    mov eax,message
    call slen
    mov edx,eax
    mov ecx,message
    mov ebx,esi
    mov eax,4
    int 80h
    mov ebx,esi
    mov eax,6
    int 80h
    mov ecx,1
    mov ebx,filename
    mov eax,5
    int 80h
    mov esi,eax
    mov edx,2
    mov ecx,0
    mov ebx,eax
    mov eax,19
    int 80h
    mov eax,name
    call slen
    mov edx,eax
    mov ecx,name
    mov ebx,esi
    mov eax,4
    int 80h
```

Рис. 2.8: Пишем программу в midnight commander

Создаем исполняемый файл и запускаем его, после этого проверяем создался ли новый файл, затем смотрим, как он заполнен (рис. 2.9).

```
[spelovandrei@fedora lab10]$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
[spelovandrei@fedora lab10]$ ld -m elf_i386 -o lab10-2 lab10-2.o
[spelovandrei@fedora lab10]$ ./lab10-2
Как Вас зовут? Андрей
[spelovandrei@fedora lab10]$ ls
in_out.asm  lab10-1.asm  lab10-1.o  lab10-2.asm  lab10-2.o  readme-1.txt
lab10-1     lab10-1.lst  lab10-2   lab10-2.lst  name.txt   readme-2.txt
[spelovandrei@fedora lab10]$ cat name.txt
Меня зовут Андрей
```

Рис. 2.9: Проверяем работу программы

3 Выводы

Мы научились писать программы для работы с файлам и научились предоставлять права доступа к файлам.