

Отчёта по лабораторной работе №10

Работа с файлами средствами Nasm.

Сидорова Арина Валерьевна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Задание для самостоятельной работы	8
3	Выводы	10

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	7
2.6	Используем команду <code>chmod</code> для установки нужных прав, после этого проверяем правильность выполнения командой <code>ls -l</code>	8
2.7	Создаем файл командой <code>touch</code>	8
2.8	Пишем программу в <code>midnight commander</code>	9
2.9	Проверяем работу программы	9

1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Выполнение лабораторной работы

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

```
avsidorova@avsidorova:~$ mkdir ~/work/arch-pc/lab10
avsidorova@avsidorova:~$ cd ~/work/arch-pc/lab10
avsidorova@avsidorova:~/work/arch-pc/lab10$ touch lab10.asm
readme-1.txt readme-2.txt
avsidorova@avsidorova:~/work/arch-pc/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. fig. 2.2).

```
/home/avsidorova/work/arch-pc/lab10/lab10.asm *
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
```

^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

Рис. 2.2: Заполняем файл

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

```
avsidorova@avsidorova:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10.lst lab10.asm
avsidorova@avsidorova:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10 lab10.o
avsidorova@avsidorova:~/work/arch-pc/lab10$ ./lab10
Введите строку для записи в файл: Hello, world!
avsidorova@avsidorova:~/work/arch-pc/lab10$
```

Рис. 2.3: Запускаем файл и проверяем его работу

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

```
avsidorova@avsidorova:~/work/arch-pc/lab10$ chmod -x lab10
avsidorova@avsidorova:~/work/arch-pc/lab10$ ./lab10
bash: ./lab10: Отказано в доступе
avsidorova@avsidorova:~/work/arch-pc/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. fig. 2.5).

```
avsidorova@avsidorova:~/work/arch-pc/lab10$ chmod +x lab10.asm
avsidorova@avsidorova:~/work/arch-pc/lab10$ ./lab10.asm
./lab10.asm: строка 1: fg: нет управления заданиями
./lab10.asm: строка 2: SECTION: команда не найдена
./lab10.asm: строка 3: filename: команда не найдена
./lab10.asm: строка 4: msg: команда не найдена
./lab10.asm: строка 5: SECTION: команда не найдена
./lab10.asm: строка 6: contents: команда не найдена
./lab10.asm: строка 7: SECTION: команда не найдена
./lab10.asm: строка 8: global: команда не найдена
./lab10.asm: строка 9: _start:: команда не найдена
./lab10.asm: строка 10: mov: команда не найдена
./lab10.asm: строка 11: call: команда не найдена
./lab10.asm: строка 12: mov: команда не найдена
./lab10.asm: строка 13: mov: команда не найдена
./lab10.asm: строка 14: call: команда не найдена
./lab10.asm: строка 15: mov: команда не найдена
./lab10.asm: строка 16: mov: команда не найдена
./lab10.asm: строка 17: mov: команда не найдена
./lab10.asm: строка 18: int: команда не найдена
./lab10.asm: строка 19: mov: команда не найдена
./lab10.asm: строка 20: mov: команда не найдена
./lab10.asm: строка 21: call: команда не найдена
./lab10.asm: строка 22: mov: команда не найдена
./lab10.asm: строка 23: mov: команда не найдена
./lab10.asm: строка 24: mov: команда не найдена
./lab10.asm: строка 25: mov: команда не найдена
```

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 13

Предоставляем права доступа к 2ум файлам, согласно варианту 20 в символьном и двоичном виде, затем проверяем работу команд. (рис. fig. 2.6).

```
авсидорова@авсидорова:~/work/arch-pc/lab10$ chmod u=w,g=x,o
= readme-1.txt
авсидорова@авсидорова:~/work/arch-pc/lab10$ chmod 631 readm
e-2.txt
авсидорова@авсидорова:~/work/arch-pc/lab10$ ls -l
итого 40
-rw-rw-r-- 1 avsidorova avsidorova 3942 окт 25 02:41 in_ou
t.asm
-rw-rw-r-- 1 avsidorova avsidorova 9732 дек 8 00:25 lab10
-rwxrwxr-x 1 avsidorova avsidorova 555 дек 8 00:23 lab10
.asm
-rw-rw-r-- 1 avsidorova avsidorova 12581 дек 8 00:25 lab10
.lst
-rw-rw-r-- 1 avsidorova avsidorova 2512 дек 8 00:25 lab10
.o
--w---x--- 1 avsidorova avsidorova 0 дек 8 00:18 readm
e-1.txt
-rw--wx--x 1 avsidorova avsidorova 0 дек 8 00:18 readm
e-2.txt
авсидорова@авсидорова:~/work/arch-pc/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. fig. 2.7).

```
авсидорова@авсидорова:~/work/arch-pc/lab10$ touch lab10-2.a
sm
авсидорова@авсидорова:~/work/arch-pc/lab10$ mc
```

Рис. 2.7: Создаем файл командой touch

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


```
/home/avsidorova/work/arch-pc/lab10/lab10-2.asm *
%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 sprintf
    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
```

^G Справка ^O Записать ^W Поиск ^K Вырезать
^X Выход ^R ЧитФайл ^\ Замена ^U Вставить

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. fig. 2.9).

```
avsidorova@avsidorova:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
avsidorova@avsidorova:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
avsidorova@avsidorova:~/work/arch-pc/lab10$ ./lab10-2
Как Вас зовут? Арина
avsidorova@avsidorova:~/work/arch-pc/lab10$ ls
in_out.asm  lab10-2.asm  lab10.asm  name.txt
lab10       lab10-2.lst  lab10.lst  readme-1.txt
lab10-2     lab10-2.o   lab10.o    readme-2.txt
avsidorova@avsidorova:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Арина
avsidorova@avsidorova:~/work/arch-pc/lab10$
```

Рис. 2.9: Проверяем работу программы

3 Выводы

Приобрели навыки написания программ для работы с файлами.