

Лабораторная работа №10

Работа с файлами средствами NASM

Хань Цзянтао

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	14
	Список литературы	15

Список иллюстраций

3.1	Ввод текста программы из листинга 10.1	8
3.2	Запуск исполняемого файла	9
3.3	Проверка выполнения программы	9
3.4	Запрет на выполнение файла	9
3.5	Добавление прав на исполнение	10
3.6	Предоставление прав доступа в символьном и <code>-x -wx rwx 000 110</code> 010двоичном виде	10
3.7	Создание и открытие файла <code>lab10-2.asm</code>	11
3.8	Написание текста программы	12
3.9	Запуск исполняемого файла и проверка его работы	13

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 Выполнение лабораторной работы

Создадим каталог для программ лабораторной работы № 10, переходим в него и создадим файлы lab10-1.asm, readme-1.txt и readme-2.txt.

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

Введём в файл lab10-1.asm текст программы, записывающей в файл сообщения, из листинга 10.1.

```

1 %include 'in_out.asm'
2 SECTION .data
3 filename db 'readme.txt', 0h ; Имя файла
4 msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
5 SECTION .bss
6 contents resb 255 ; переменная для вводимой строки
7 SECTION .text
8 global _start
9 _start:
10 ; --- Печать сообщения 'msg'
11 mov eax, msg
12 call sprint
13 ; ---- Запись введенной с клавиатуры строки в 'contents'
14 mov ecx, contents
15 mov edx, 255
16 call sread
17 ; --- Открытие существующего файла ('sys_open')
18 mov ecx, 2 ; открываем для записи (2)
19 mov ebx, filename
20 mov eax, 5
21 int 80h
22 ; --- Запись дескриптора файла в 'esi'
23 mov esi, eax
24 ; --- Расчет длины введенной строки
25 mov eax, contents ; в 'eax' запишется количество
26 call slen ; введенных байтов
27 ; --- Записываем в файл 'contents' ('sys_write')
28 mov edx, eax
29 mov ecx, contents
30 mov ebx, esi
31 mov eax, 4
32 int 80h
33 ; --- Закрываем файл ('sys_close')
34 mov ebx, esi
35 mov eax, 6
36 int 80h
37 call quit

```

Рис. 3.1: Ввод текста программы из листинга 10.1

Создадим исполняемый файл и проверю его работу.

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: AAprispeshkin
```

Рис. 3.2: Запуск исполняемого файла

Проверим правильность выполнения программы

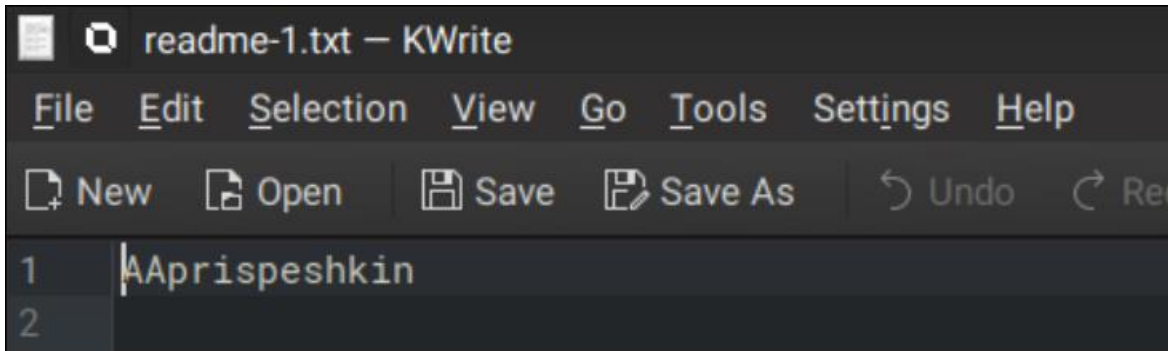


Рис. 3.3: Проверка выполнения программы

Далее с помощью команды `chmod u-x` изменим права доступа к исполняемому файлу `lab10-1`, запретив его выполнение и попытаемся выполнить файл.

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ chmod u-x lab10-1
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Permission denied
```

Рис. 3.4: Запрет на выполнение файла

Файл не выполняется, т.к в команде мы указали “u” - владелец, “-” - отменить набор прав, “x” - право на исполнение.

С помощью команды `chmod u+x` изменим права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение, и попытаемся выполнить его.

```

aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ chmod u+x lab10-1.asm
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: line 1: fg: no job control
./lab10-1.asm: line 2: SECTION: command not found
./lab10-1.asm: line 3: filename: command not found
./lab10-1.asm: line 3: Имя: command not found
./lab10-1.asm: line 4: msg: command not found
./lab10-1.asm: line 4: Сообщение: command not found
./lab10-1.asm: line 5: SECTION: command not found
./lab10-1.asm: line 6: contents: command not found
./lab10-1.asm: line 6: переменная: command not found
./lab10-1.asm: line 7: SECTION: command not found
./lab10-1.asm: line 8: global: command not found
./lab10-1.asm: line 9: _start:: command not found
./lab10-1.asm: line 10: syntax error near unexpected token `;'
./lab10-1.asm: line 10: `; --- Печать сообщения `msg`'

```

Рис. 3.5: 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 (1) в таблице 10.4 предоставляем права доступа к файлу readme1.txt представленные в символьном виде, а для файла readme-2.txt – в двоичном виде:

–x -wx rwx 000 110 010

И проверим правильность выполнения с помощью команды ls -l.

```

aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ chmod 640 readme-1.txt # --x -wx rwx
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ chmod 640 readme-2.txt # 000 110 010
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ls -l
total 44
-rw-r--r-- 1 aaprispeshkin aaprispeshkin 3942 ноя  8 16:08 in_out.asm
-rwxr-xr-x 1 aaprispeshkin aaprispeshkin 9744 дек 16 16:45 lab10-1
-rwxr--r-- 1 aaprispeshkin aaprispeshkin 1142 дек 16 16:35 lab10-1.asm
-rw-r--r-- 1 aaprispeshkin aaprispeshkin 13448 дек 16 16:44 lab10-1.lst
-rw-r--r-- 1 aaprispeshkin aaprispeshkin 2528 дек 16 16:44 lab10-1.o
-rw-r----- 1 aaprispeshkin aaprispeshkin 14 дек 16 18:21 readme-1.txt
-rw-r----- 1 aaprispeshkin aaprispeshkin 0 дек 14 13:42 readme-2.txt

```

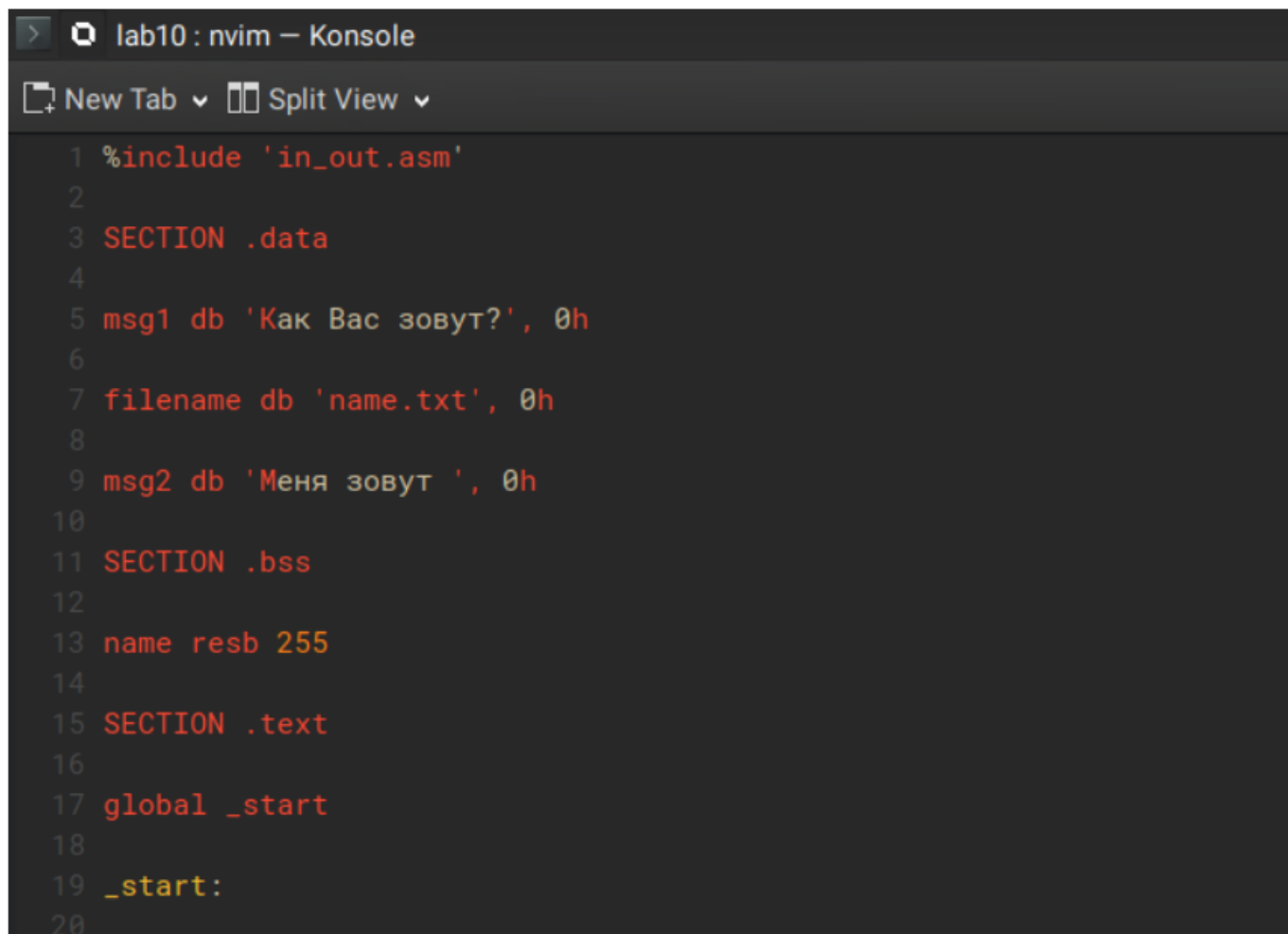
Рис. 3.6: Предоставление прав доступа в символьном и –x -wx rwx 000 110 010 двоичном виде

Создадим файл lab10-2.asm и откроем его в текстовом редакторе neovim

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ touch lab10-2.asm  
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ nvim lab10-2.asm
```

Рис. 3.7: Создание и открытие файла lab10-2.asm

Запишем код программы, выводящей приглашения “Как Вас зовут?”, считывающей с клавиатуры фамилию и имя и создающую файл, в который записывается сообщение “Меня зовут:”.

A screenshot of a terminal window titled 'lab10 : nvim — Konsole'. The window shows a text editor with assembly code. The code is as follows:

```
1 %include 'in_out.asm'
2
3 SECTION .data
4
5 msg1 db 'Как Вас зовут?', 0h
6
7 filename db 'name.txt', 0h
8
9 msg2 db 'Меня зовут ', 0h
10
11 SECTION .bss
12
13 name resb 255
14
15 SECTION .text
16
17 global _start
18
19 _start:
20
```

Рис. 3.8: Написание текста программы

Создадим исполняемый файл и проверим его работу. Проверим наличие файла и его содержимое с помощью команд `ls` и `cat`.

```
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
aaprispeshkin:[aaprispeshkin]:~/work/arch-pc/lab10$ ./lab10-2
```

Рис. 3.9: Запуск исполняемого файла и проверка его работы

4 Выводы

В результате данной лабораторной работы я научился работать с файлами средствами языка программирования NASM

Список литературы

Лабораторная работа №10