

Отчёт по лабораторной работе №10

Работа с файлами средствами Nasm

Лань Цяньин

1. Цель работы

Приобретение навыков написания программ для работы с файлами

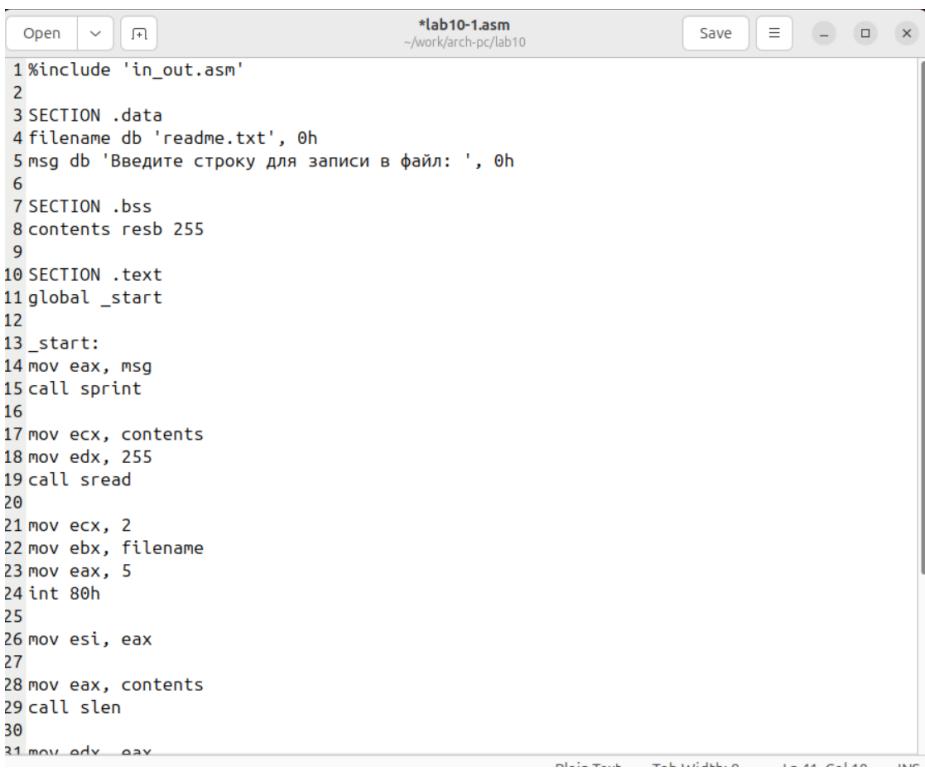
2. Порядок выполнения лабораторной работы

На данном этапе выполняется подготовка рабочего окружения для лабораторной работы №10. Последовательно создаются необходимые файлы, копируется библиотечный модуль и открывается редактор для написания исходного кода.(рис. Рисунок 1)

```
clanj1@clanj1:~/work/arch-pc/lab10$ cd ~/work/arch-pc/lab10
clanj1@clanj1:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
clanj1@clanj1:~/work/arch-pc/lab10$ cp /media/sf_VB_Share/in_out.asm ~/work/arch-pc/lab10
clanj1@clanj1:~/work/arch-pc/lab10$ gedit lab10-1.asm
```

Рисунок 1: Подготовка файлов для lab10

На рисунке показан исходный код программы lab10-1.asm из листинга 10.1. (рис. Рисунок 2)



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 filename db 'readme.txt', 0h
5 msg db 'Введите строку для записи в файл: ', 0h
6
7 SECTION .bss
8 contents resb 255
9
10 SECTION .text
11 global _start
12
13 _start:
14 mov eax, msg
15 call sprint
16
17 mov ecx, contents
18 mov edx, 255
19 call sread
20
21 mov ecx, 2
22 mov ebx, filename
23 mov eax, 5
24 int 80h
25
26 mov esi, eax
27
28 mov eax, contents
29 call slen
30
31 mov edx, eax
```

Рисунок 2: Исходный код программы lab10-1.asm

Программа была успешно скомпилирована и скомпонована. В ходе тестирования через стандартный ввод была передана строка «999999». После выполнения программы созданный файл readme.txt был проверен командой cat, которая вывела введённую строку. Результат подтверждает корректность реализации файлового вывода в программе(рис. Рисунок 3)

```
clanj1@clanj1:~/work/arch-pc/lab10$ nasm -f elf lab10-1.asm
clanj1@clanj1:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: 999999
clanj1@clanj1:~/work/arch-pc/lab10$ cat readme-1.txt readme-2.txt
clanj1@clanj1:~/work/arch-pc/lab10$ touch readme.txt
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: 999999
clanj1@clanj1:~/work/arch-pc/lab10$ cat readme.txt
999999
```

Рисунок 3: Этапы сборки и тестирования программы lab10-1

На рисунке последовательно показаны ключевые этапы работы с програм-

мой lab10-1.asm: компиляция, компоновка, запуск и проверка результата. Программа считывает строку и записывает её в указанный файл.

После выполнения команды chmod -x lab10-1, которая отнимает право на выполнение у файла, попытка запуска (./lab10-1) завершается ошибкой Permission denied.(рис. Рисунок 4)

```
clanj1@clanj1:~/work/arch-pc/lab10$ chmod -x lab10-1
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Permission denied
```

Рисунок 4: Результат изменения прав доступа к файлу lab10-1

На рисунке показан результат изменения прав доступа к исполняемому файлу с помощью команды chmod и попытка его запуска.

Команда chmod +x добавила файлу права на выполнение, но запустить его не удалось: система попыталась интерпретировать ассемблерный код как команды оболочки, что привело к ошибкам. Это показывает, что для выполнения файл должен содержать машинный код, а не исходный текст. (рис. Рисунок 5)

```
clanj1@clanj1:~/work/arch-pc/lab10$ chmod +x lab10-1.asm
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: line 1: fg: no job control
./lab10-1.asm: line 3: SECTION: command not found
./lab10-1.asm: line 4: filename: command not found
./lab10-1.asm: line 5: msg: command not found
./lab10-1.asm: line 7: SECTION: command not found
./lab10-1.asm: line 8: contents: command not found
./lab10-1.asm: line 10: SECTION: command not found
./lab10-1.asm: line 11: global: command not found
./lab10-1.asm: line 13: _start:: command not found
./lab10-1.asm: line 14: mov: command not found
./lab10-1.asm: line 15: call: command not found
./lab10-1.asm: line 17: mov: command not found
./lab10-1.asm: line 18: mov: command not found
./lab10-1.asm: line 19: call: command not found
./lab10-1.asm: line 21: mov: command not found
./lab10-1.asm: line 22: mov: command not found
./lab10-1.asm: line 23: mov: command not found
./lab10-1.asm: line 24: int: command not found
./lab10-1.asm: line 26: mov: command not found
./lab10-1.asm: line 28: mov: command not found
[1]+ 10- Stopped                 ./lab10-1
```

Рисунок 5: Попытка выполнения файла с исходным кодом

На рисунке показана попытка запуска на выполнение файла с исходным кодом на ассемблере после добавления ему соответствующих прав доступа.

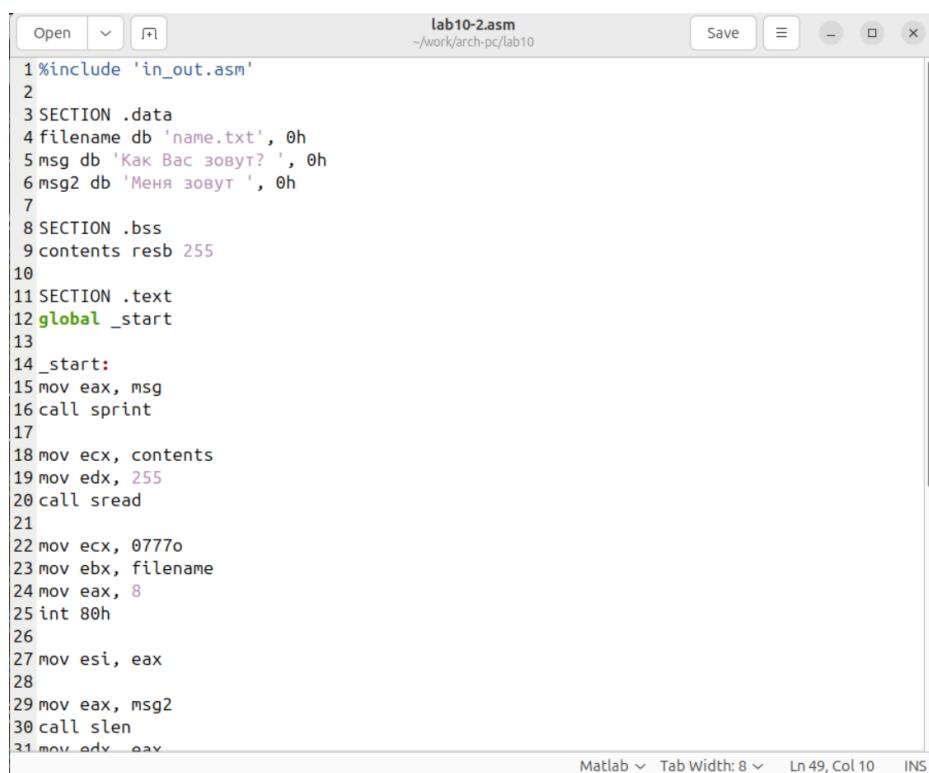
В ходе выполнения задания для моего варианта №9 были установлены права доступа для файлов `readme-1.txt` и `readme-2.txt`.(рис. Рисунок 6)

```
clanj1@clanj1:~/work/arch-pc/lab10$ chmod u=x,g=w,o=w readme-1.txt
clanj1@clanj1:~/work/arch-pc/lab10$ chmod 135 readme-2.txt
clanj1@clanj1:~/work/arch-pc/lab10$ ls -l readme-1.txt readme-2.txt
--x-w--w- 1 clanj1 clanj1 0 Dec 23 15:09 readme-1.txt
---x-wx--x 1 clanj1 clanj1 0 Dec 23 15:09 readme-2.txt
```

Рисунок 6: Установка прав доступа для файлов (вариант 9)

В результате для файла `readme-1.txt` (команда `chmod u=x,g=w,o=w`) установлены права `--x-w--w-`, а для `readme-2.txt` (команда `chmod 135`) — права `--wxr-x`. Вывод команды `ls -l` подтверждает точное соответствие между применёнными командами и итоговыми правами,

3. Задание для самостоятельной работы



The screenshot shows a text editor window with the following assembly code:

```
1 %include 'in_out.asm'
2
3 SECTION .data
4 filename db 'name.txt', 0h
5 msg db 'Как Вас зовут? ', 0h
6 msg2 db 'Меня зовут ', 0h
7
8 SECTION .bss
9 contents resb 255
10
11 SECTION .text
12 global _start
13
14 _start:
15 mov eax, msg
16 call sprint
17
18 mov ecx, contents
19 mov edx, 255
20 call sread
21
22 mov ecx, 07770
23 mov ebx, filename
24 mov eax, 8
25 int 80h
26
27 mov esi, eax
28
29 mov eax, msg2
30 call slen
31 mov edx, eax
```

The code is an assembly program named lab10-2.asm. It includes an include directive for 'in_out.asm'. The .data section contains variable definitions for 'filename' (name.txt), 'msg' ('Как Вас зовут?'), and 'msg2' ('Меня зовут'). The .bss section defines a buffer 'contents' of size 255 bytes. The .text section contains the entry point '_start' which prints 'msg' using 'sprint', reads input from the user using 'sread' (with a buffer size of 07770h), and then prints 'msg2' followed by the user's input using 'slen'.

Рисунок 1: Исходный код программы lab10-2.asm

На рисунке представлен исходный код программы lab10-2.asm. (рис. Рисунок 1)

После сборки исполняемого файла программа была запущена. Она вывела приглашение ввести имя, на что был дан ответ «Лань Цяньин». Затем с помо-

щью команды `cat` было проверено содержимое созданного программой файла `name.txt`. Файл содержал ожидаемую строку «Меня зовут Лань Цяньин», что подтверждает корректную работу всех этапов программы: ввода данных, создания файла и записи в него.(рис. Рисунок 2)

```
clanj1@clanj1:~/work/arch-pc/lab10$ nasm -f elf lab10-2.asm
clanj1@clanj1:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-2.asm
bash: ./Lab10-2.asm: Permission denied
clanj1@clanj1:~/work/arch-pc/lab10$ ./lab10-2
Как Вас зовут? Лань Цяньин
clanj1@clanj1:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Лань Цяньин
```

Рисунок 2: Запуск и проверка программы `lab10-2`

Программа `lab10-2.asm` была успешно откомпилирована и запущена.

4 ВЫВОД

В лабораторной работе научился работать с файлами через системные вызовы NASM. На практике использовал sys_creat для создания файла и sys_write для записи данных, а также команду chmod для управления правами доступа. Результат — программа, которая запрашивает имя и сохраняет его в файл. Работа выполнена.