

Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Арфонос Дмитрий

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Самостоятельная работа	10
4	Вывод	15

Список иллюстраций

2.1	Создание директории	5
2.2	Редактирование текста	6
2.3	Запуск исполняемого файла	7
2.4	Проверка работы программы	7
2.5	Доступ к файлу	8
2.6	Изменение прав к файлу	8
2.7	Предоставляю права доступа к файлу	9
2.8	Предоставляю права доступа к файлу	9
3.1	Создание файла для самостоятельной	10
3.2	Написание программы для самостоятельной	11
3.3	Исполнение файла	11
3.4	вставляю программу	12

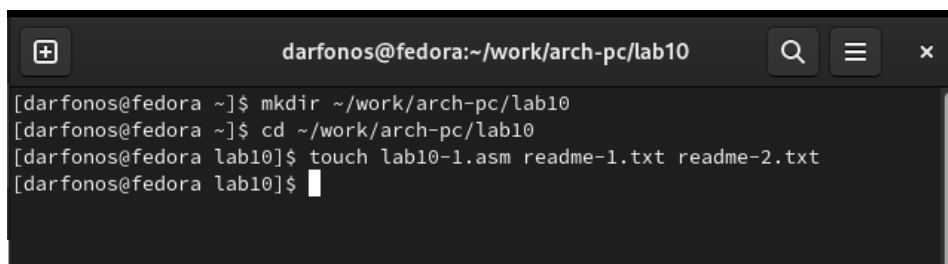
1 Цель работы

Приобретение навыков написания программ для работы с файлами в NASM.

2 Выполнение лабораторной работы

Шаг 1

С помощью утилиты `mkdir` создаю директорию `lab10`, перехожу в нее и создаю файл для работы. (рис. [2.1])

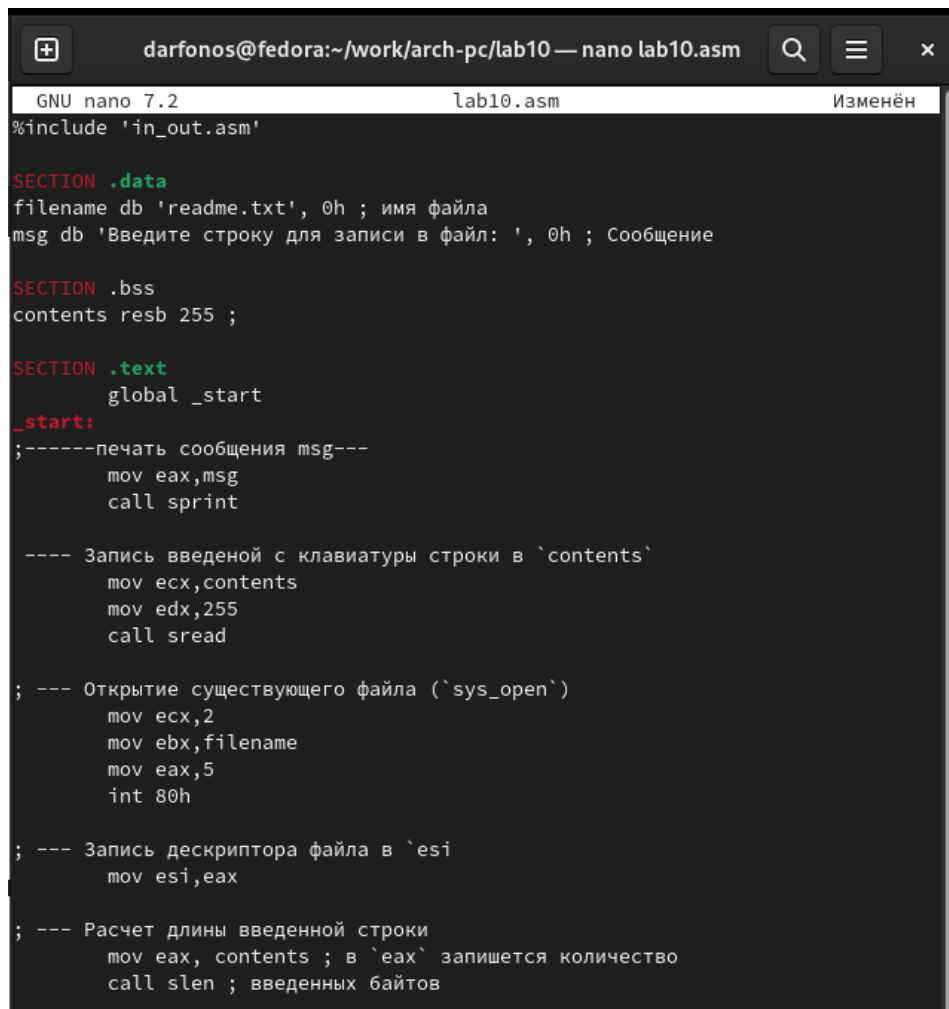


```
darfonos@fedora:~/work/arch-pc/lab10
[darfonos@fedora ~]$ mkdir ~/work/arch-pc/lab10
[darfonos@fedora ~]$ cd ~/work/arch-pc/lab10
[darfonos@fedora lab10]$ touch lab10-1.asm readme-1.txt readme-2.txt
[darfonos@fedora lab10]$
```

Рис. 2.1: Создание директории

Шаг 2

Открываю созданный файл `lab10-1.asm`, вставляю в него программу из Листинга.(рис. [2.2]).



```
darfonos@fedora:~/work/arch-pc/lab10 — nano lab10.asm
GNU nano 7.2 lab10.asm Изменён
%include 'in_out.asm'

SECTION .data
filename db 'readme.txt', 0h ; имя файла
msg db 'Введите строку для записи в файл: ', 0h ; Сообщение

SECTION .bss
contents resb 255 ;

SECTION .text
global _start
_start:
;-----печать сообщения msg---
    mov eax,msg
    call sprint

    ---- Запись введенной с клавиатуры строки в `contents`
    mov ecx,contents
    mov edx,255
    call sread

; --- Открытие существующего файла (`sys_open`)
    mov ecx,2
    mov ebx,filename
    mov eax,5
    int 80h

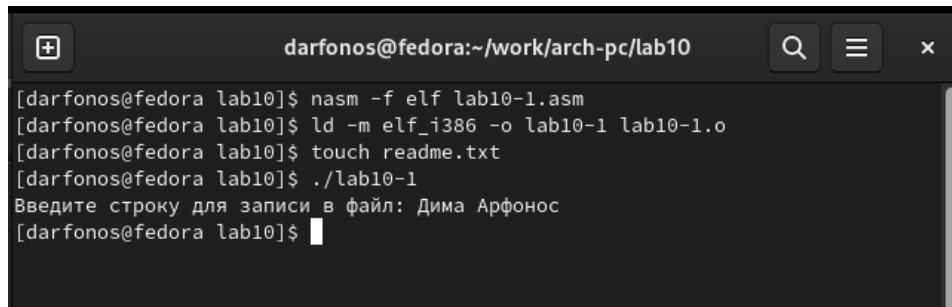
; --- Запись дескриптора файла в `esi`
    mov esi,eax

; --- Расчет длины введенной строки
    mov eax, contents ; в `eax` запишется количество
    call slen ; введенных байтов
```

Рис. 2.2: Редактирование текста

Шаг 3

Создаю исполняемый файл программы, а также сам файл, куда будет записываться строка. (рис. [2.3]).

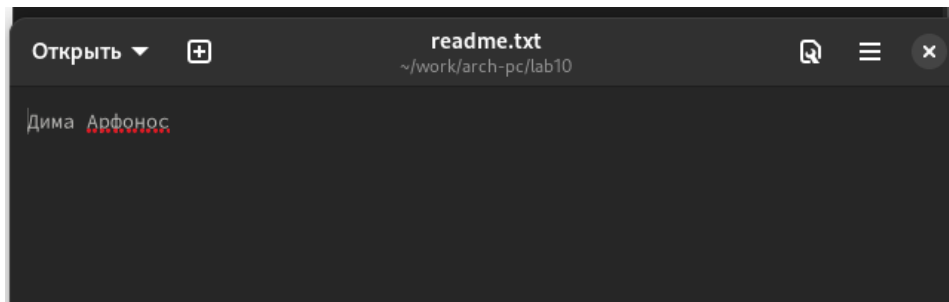


```
darfonos@fedora:~/work/arch-pc/lab10
[darfonos@fedora lab10]$ nasm -f elf lab10-1.asm
[darfonos@fedora lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[darfonos@fedora lab10]$ touch readme.txt
[darfonos@fedora lab10]$ ./lab10-1
Введите строку для записи в файл: Дима Арфонов
[darfonos@fedora lab10]$
```

Рис. 2.3: Запуск исполняемого файла

Шаг 4

Проверяю работу программы. (рис. [2.4]).



```
Открыть ▾ + readme.txt
~/work/arch-pc/lab10
Дима Арфонов
```

Рис. 2.4: Проверка работы программы

Шаг 5

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. (рис. [2.5]).

```
darfonos@fedora:~/work/arch-pc/lab10
[darfonos@fedora lab10]$ chmod 640 lab10-1
[darfonos@fedora lab10]$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
[darfonos@fedora lab10]$
```

Рис. 2.5: Доступ к файлу

1. В символьном представлении есть возможность явно указывать какой группе какие права необходимо добавить или отнять. Так '6' дает право владельцу лишь для чтения и записи, но не для исполнения.
2. '4' дает право группе лишь для чтения.
3. '0' не отнимает все права для всех прочих пользователей

Шаг 6

С помощью команды `chmod` изменяю права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. (рис. [2.6]).

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

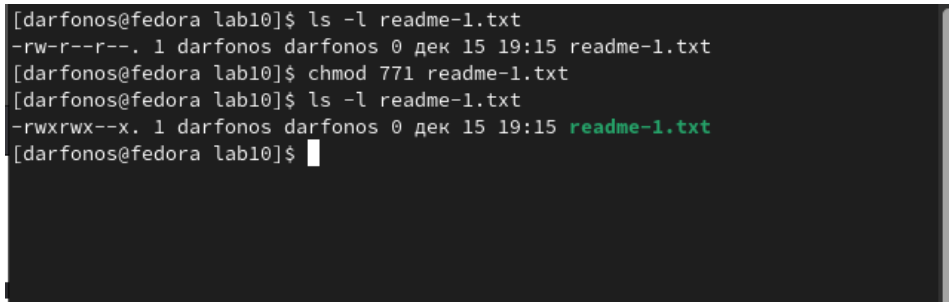
Рис. 2.6: Изменение прав к файлу

- С помощью аргументов "u" и "+" добавляем право владельцу на исполнение файла, однако в файле находится исходный код программы, который еще не ассемблирован, и его выполнение невозможно.

- Происходит исполнение файла, но она не будет работать.
-

Шаг 7

В соответствии со своим 2-ым вариантом в таблице, предоставляю права доступа к файлу `readme-1.txt` представленные в символьном виде: `rw-rwx-rwx` (рис. [2.7]).



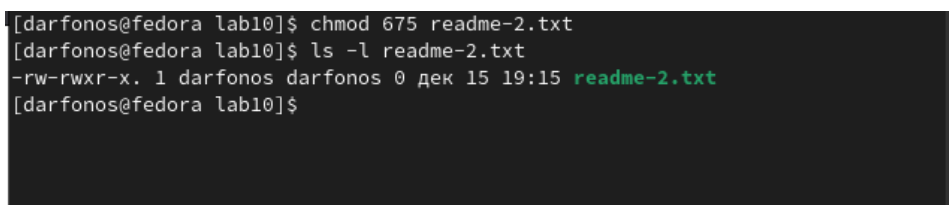
```
[darfonos@fedora lab10]$ ls -l readme-1.txt
-rw-r--r--. 1 darfonos darfonos 0 дек 15 19:15 readme-1.txt
[darfonos@fedora lab10]$ chmod 771 readme-1.txt
[darfonos@fedora lab10]$ ls -l readme-1.txt
-rwxrwx--x. 1 darfonos darfonos 0 дек 15 19:15 readme-1.txt
[darfonos@fedora lab10]$
```

Рис. 2.7: Предоставляю права доступа к файлу

- Вначале проверяю изначальные права, которые имеют пользователи к файлу по умолчанию, затем даю ей другие права в соответствии с вариантом.
-

Шаг 8

Предоставляю права доступа к файлу `readme-2.txt` представленные в символьном – в двоичном виде: `110 111 101`, то есть `675` (рис. [2.8]).



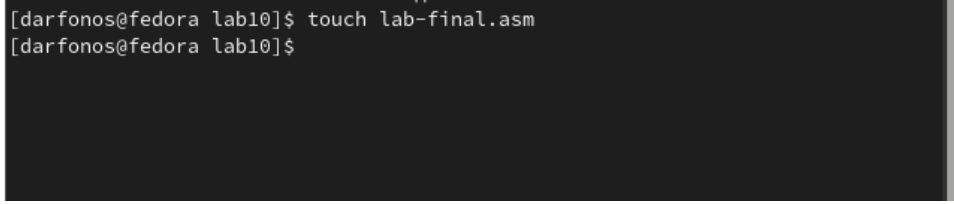
```
[darfonos@fedora lab10]$ chmod 675 readme-2.txt
[darfonos@fedora lab10]$ ls -l readme-2.txt
-rw-rwxr-x. 1 darfonos darfonos 0 дек 15 19:15 readme-2.txt
[darfonos@fedora lab10]$
```

Рис. 2.8: Предоставляю права доступа к файлу

3 Самостоятельная работа

Шаг 1

Создаю новый файл для самостоятельной работы(рис.[3.1]).

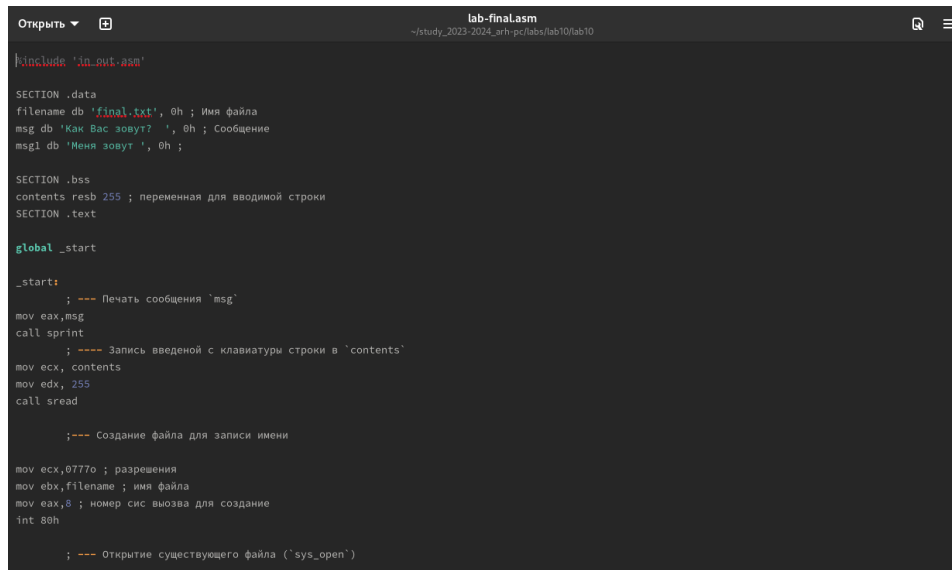


```
[darfonos@fedora lab10]$ touch lab-final.asm  
[darfonos@fedora lab10]$
```

Рис. 3.1: Создание файла для самостоятельной

Шаг 2

Пишу в неё текст программы, которая будет запрашивать имя пользователя, будет создавать файл и записывать в нее введенное имя, добавив в начале текст : “Меня зовут”. (рис. [3.2]).



```
Открыть ▾ [icon] lab-final.asm
~/study_2023-2024_arh/pc/labs/lab10/lab10

;используем 'lab-final.asm'

SECTION .data
filename db 'final.txt', 0h ; имя файла
msg db 'Как Вас зовут? ', 0h ; Сообщение
msg1 db 'Меня зовут ', 0h ;

SECTION .bss
contents resb 255 ; переменная для вводимой строки
SECTION .text

global _start

_start:
; --- Печать сообщения 'msg'
mov eax, msg
call sprint
; ---- Запись введенной с клавиатуры строки в 'contents'
mov ecx, contents
mov edx, 255
call sread

; --- Создание файла для записи имени

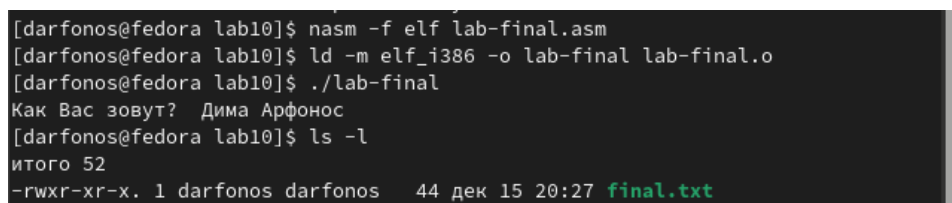
mov ecx, 0777o ; разрешения
mov ebx, filename ; имя файла
mov eax, 8 ; номер сис. вызова для создание
int 80h

; --- Открытие существующего файла ('sys_open')
```

Рис. 3.2: Написание программы для самостоятельной

Шаг 3

- Создаю исполняемый файл и запускаю её, ввожу свое имя и фамилию. (рис. [3.3]).
- Сразу проверяю с помощью атрибута **ls** созданся ли необходимый файл, куда должны записываться введенные символы.



```
[darfonos@fedora lab10]$ nasm -f elf lab-final.asm
[darfonos@fedora lab10]$ ld -m elf_i386 -o lab-final lab-final.o
[darfonos@fedora lab10]$ ./lab-final
Как Вас зовут? Дима Арфонос
[darfonos@fedora lab10]$ ls -l
итого 52
-rwxr-xr-x. 1 darfonos darfonos  44 дек 15 20:27 final.txt
```

Рис. 3.3: Исполнение файла

Шаг 4

С помощью атрибуты **cat** проверяю содержимое созданного файла. (рис. [3.4]).

```
-rwxrwx--x. 1 darfonos darfonos    0 дек 15 19:15 readme-1.txt
-rw-rwxr-x. 1 darfonos darfonos    0 дек 15 19:15 readme-2.txt
-rw-r--r--. 1 darfonos darfonos   24 дек 15 19:50 readme.txt
[darfonos@fedora lab10]$ cat final.txt
Меня зовут Дима Арфонос
[darfonos@fedora lab10]$
```

Рис. 3.4: вставляю программу

Программа отработала корректно!!

Текст программы в самостоятельной работе

```
%include 'in_out.asm'
```

```
SECTION .data
```

```
filename db 'final.txt', 0h ; Имя файла
```

```
msg db 'Как Вас зовут? ', 0h ; Сообщение
```

```
msg1 db 'Меня зовут ', 0h ;
```

```
SECTION .bss
```

```
contents resb 255 ; переменная для вводимой строки
```

```
SECTION .text
```

```
global _start
```

```
_start:
```

```
    ; --- Печать сообщения `msg`
```

```
mov eax,msg
```

```
call sprint
```

```

        ; ---- Запись введенной с клавиатуры строки в `contents`
mov ecx, contents
mov edx, 255
call sread

        ; --- Создание файла для записи имени

mov ecx, 0777o ; разрешения
mov ebx, filename ; имя файла
mov eax, 8 ; номер сис вызова для создание
int 80h

        ; --- Открытие существующего файла (`sys_open`)

mov ecx, 2 ; открываем для записи (2)
mov ebx, filename
mov eax, 5
int 80h

        ; --- Запись дескриптора файла в `esi`

mov esi, eax

        ; --- Расчет длины введенной строки msg1
mov eax, msg1
call slen

        ; --- Записываем в файл `msg1` (`sys_write`)
mov edx, eax
mov ecx, msg1

```

```
mov ebx, esi
mov eax, 4
int 80h

; --- Расчет длины введенной строки
mov eax, contents ; в `eax` запишется количество
call slen ; введенных байтов

; --- Записываем в файл `contents` (`sys_write`)
mov edx, eax
mov ecx, contents
mov ebx, esi
mov eax, 4
int 80h

; --- Закрываем файл (`sys_close`)
mov ebx, esi
mov eax, 6
int 80h

call quit
```

4 Вывод

В ходе выполнения работы были получены навыки по работе с файлами в NASM.