

Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Бондаренко Кристина

Содержание

1	Цель работы	5
2	Лабораторная работа	6
3	Самостоятельная работа	9
3.1	Программа отработала корректно!!	11
3.2	Текст программы в самостоятельной работе	11
4	Выводы	14

Список иллюстраций

2.1	Создание директории	6
2.2	Редактирование текста	7
2.3	Запуск исполняемого файла	7
2.4	Доступ к файлу	8
2.5	Предоставляю права доступа к файлу	8
2.6	Предоставляю права доступа к файлу	8
3.1	Создание файла для самостоятельной работы	9
3.2	Написание программы для самостоятельной работы	10
3.3	Исполнение файла	11

Список таблиц

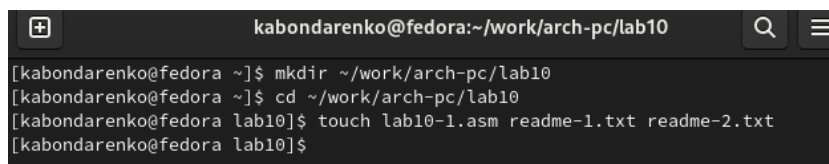
1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Лабораторная работа

Шаг 1

С помощью утилиты `mkdir` создаю директорию `lab10`, перехожу в нее и создаю файл для работы.

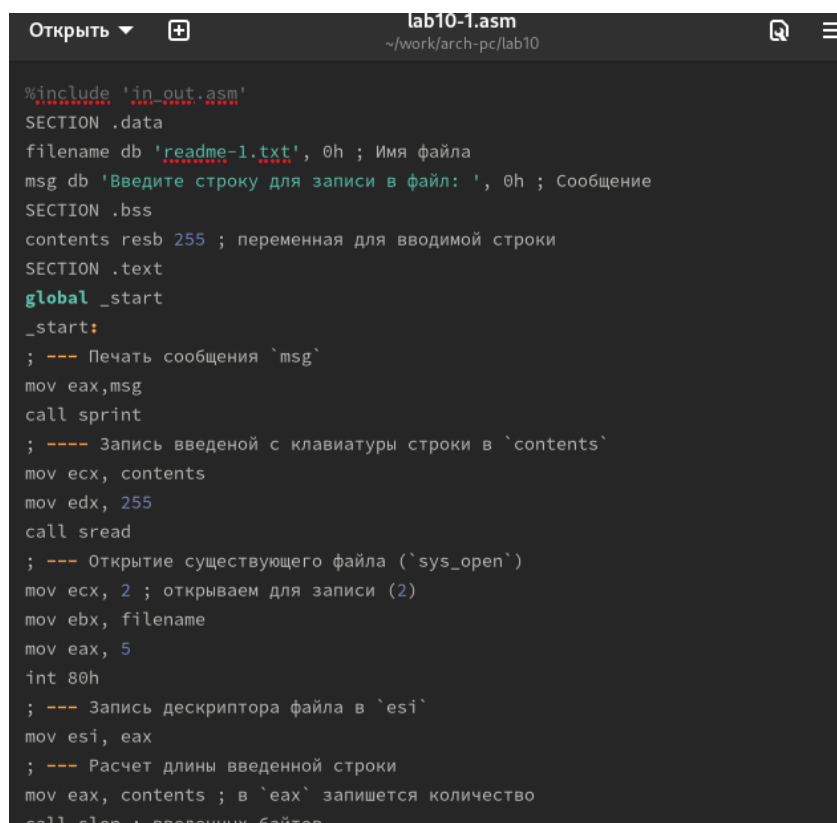


```
kabondarenko@fedora:~/work/arch-pc/lab10
[kabondarenko@fedora ~]$ mkdir ~/work/arch-pc/lab10
[kabondarenko@fedora ~]$ cd ~/work/arch-pc/lab10
[kabondarenko@fedora lab10]$ touch lab10-1.asm readme-1.txt readme-2.txt
[kabondarenko@fedora lab10]$
```

Рис. 2.1: Создание директории

Шаг 2

Открываю созданный файл `lab10-1.asm`, вставляю в него программу из Листинга



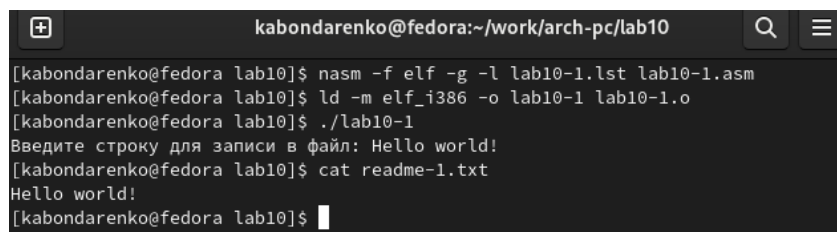
```
lab10-1.asm
~/work/arch-pc/lab10

%include 'in_out.asm'
SECTION .data
filename db 'readme-1.txt', 0h ; Имя файла
msg db 'Введите строку для записи в файл: ', 0h ; Сообщение
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
; --- 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
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.



```
kabondarenko@fedora:~/work/arch-pc/lab10

[kabondarenko@fedora lab10]$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
[kabondarenko@fedora lab10]$ ld -m elf_i386 -o lab10-1 lab10-1.o
[kabondarenko@fedora lab10]$ ./lab10-1
Введите строку для записи в файл: Hello world!
[kabondarenko@fedora lab10]$ cat readme-1.txt
Hello world!
[kabondarenko@fedora lab10]$
```

Рис. 2.3: Запуск исполняемого файла

Шаг 4

С помощью команды `chmod` изменяю права доступа к исполняемому файлу `lab10-1`, запретив его выполнение.

```
kabondarenko@fedora:~/work/arch-pc/lab10
[kabondarenko@fedora lab10]$ chmod 152 lab10-1
[kabondarenko@fedora lab10]$ cat lab10-1
cat: lab10-1: Отказано в доступе
[kabondarenko@fedora lab10]$
```

Рис. 2.4: Доступ к файлу

Шаг 5

В соответствии со своим 5-ым вариантом в таблице, предоставляю права доступа к файлу `readme-1.txt` представленные в символьном виде: `-x -w- r-x`

```
kabondarenko@fedora:~/work/arch-pc/lab10
[kabondarenko@fedora lab10]$ chmod go+w readme-1.txt
[kabondarenko@fedora lab10]$ ls -l readme-1.txt
-rw-rw-rw-. 1 kabondarenko dayanchberdyev 13 фев 13 17:16 readme-1.txt
[kabondarenko@fedora lab10]$ chmod 152 readme-2.txt
[kabondarenko@fedora lab10]$ ls -l readme-2.txt
---xr-x-w-. 1 kabondarenko dayanchberdyev 0 фев 13 17:14 readme-2.txt
[kabondarenko@fedora lab10]$
```

Рис. 2.5: Предоставляю права доступа к файлу

Шаг 6

Предоставляю права доступа к файлу `readme-2.txt` представленные в символьном – в двочном виде: `001 101 010`, то есть `1 5 2`

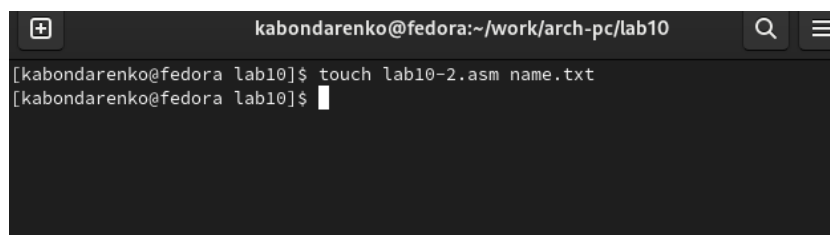
```
kabondarenko@fedora:~/work/arch-pc/lab10
[kabondarenko@fedora lab10]$ chmod go+w readme-1.txt
[kabondarenko@fedora lab10]$ ls -l readme-1.txt
-rw-rw-rw-. 1 kabondarenko dayanchberdyev 13 фев 13 17:16 readme-1.txt
[kabondarenko@fedora lab10]$ chmod 152 readme-2.txt
[kabondarenko@fedora lab10]$ ls -l readme-2.txt
---xr-x-w-. 1 kabondarenko dayanchberdyev 0 фев 13 17:14 readme-2.txt
[kabondarenko@fedora lab10]$
```

Рис. 2.6: Предоставляю права доступа к файлу

3 Самостоятельная работа

Шаг 1

Создаю новый файлы для самостоятельной работы

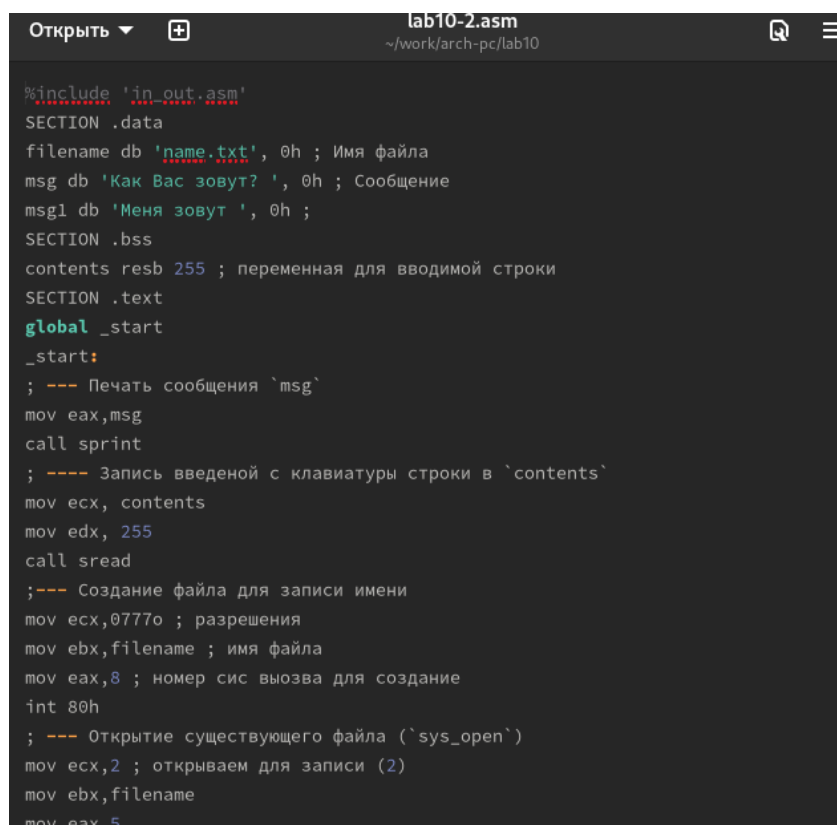
A terminal window with a dark background. The title bar shows 'kabondarenko@fedora:~/work/arch-pc/lab10'. The prompt is '[kabondarenko@fedora lab10]\$'. The command 'touch lab10-2.asm name.txt' has been entered and executed. The prompt is now '[kabondarenko@fedora lab10]\$' with a cursor.

```
kabondarenko@fedora:~/work/arch-pc/lab10
[kabondarenko@fedora lab10]$ touch lab10-2.asm name.txt
[kabondarenko@fedora lab10]$
```

Рис. 3.1: Создание файла для самостоятельной работы

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”.



```
lab10-2.asm
~/work/arch-pc/lab10

%include 'in_out.asm'
SECTION .data
filename db 'name.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
```

Рис. 3.2: Написание программы для самостоятельной работы

Шаг 3

Создаю исполняемый файл и запускаю её, ввожу свое имя и фамилию. С помощью атрибуты cat проверяю содержимое созданного файла.

```

[kabondarenko@fedora lab10]$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
[kabondarenko@fedora lab10]$ ld -m elf_i386 -o lab10-2 lab10-2.o
[kabondarenko@fedora lab10]$ ./lab10-2
Как Вас зовут? Бондаренко Кристина
[kabondarenko@fedora lab10]$ ls -l
итого 84
-rw-r--r--. 1 kabondarenko dayanchberdyev 3942 ноя 12 22:43 in_out.asm
---xr-x-w-. 1 kabondarenko dayanchberdyev 9744 фев 13 17:16 lab10-1
-rw-r--r--. 1 kabondarenko dayanchberdyev 1142 дек 15 21:58 lab10-1.asm
-rw-r--r--. 1 kabondarenko dayanchberdyev 13448 фев 13 17:15 lab10-1.lst
-rw-r--r--. 1 kabondarenko dayanchberdyev 2528 фев 13 17:15 lab10-1.o
-rwxr-xr-x. 1 kabondarenko dayanchberdyev 9760 фев 13 17:23 lab10-2
-rw-r--r--. 1 kabondarenko dayanchberdyev 1559 дек 16 20:11 lab10-2.asm
-rw-r--r--. 1 kabondarenko dayanchberdyev 14397 фев 13 17:22 lab10-2.lst
-rw-r--r--. 1 kabondarenko dayanchberdyev 2608 фев 13 17:22 lab10-2.o
-rw-r--r--. 1 kabondarenko dayanchberdyev 58 фев 13 17:23 name.txt
-rw-rw-rw-. 1 kabondarenko dayanchberdyev 13 фев 13 17:16 readme-1.txt
---xr-x-w-. 1 kabondarenko dayanchberdyev 0 фев 13 17:14 readme-2.txt
[kabondarenko@fedora lab10]$ cat name.txt
Меня зовут Бондаренко Кристина
[kabondarenko@fedora lab10]$

```

Рис. 3.3: Исполнение файла

3.1 Программа отработала корректно!!

3.2 Текст программы в самостоятельной работе

```

#include 'in_out.asm'

SECTION .data
filename db 'name.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 Выводы

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