

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

дисциплина: Архитектура компьютера

Студент: Тимаков Макарий Владимирович

Группа: НПИбд 02-25

МОСКВА

2025г.

1. Цель работы

Приобрести навыки написания программ для работы с файлами, научиться управлять доступом к файлам.

2. Выполнение лабораторной работы

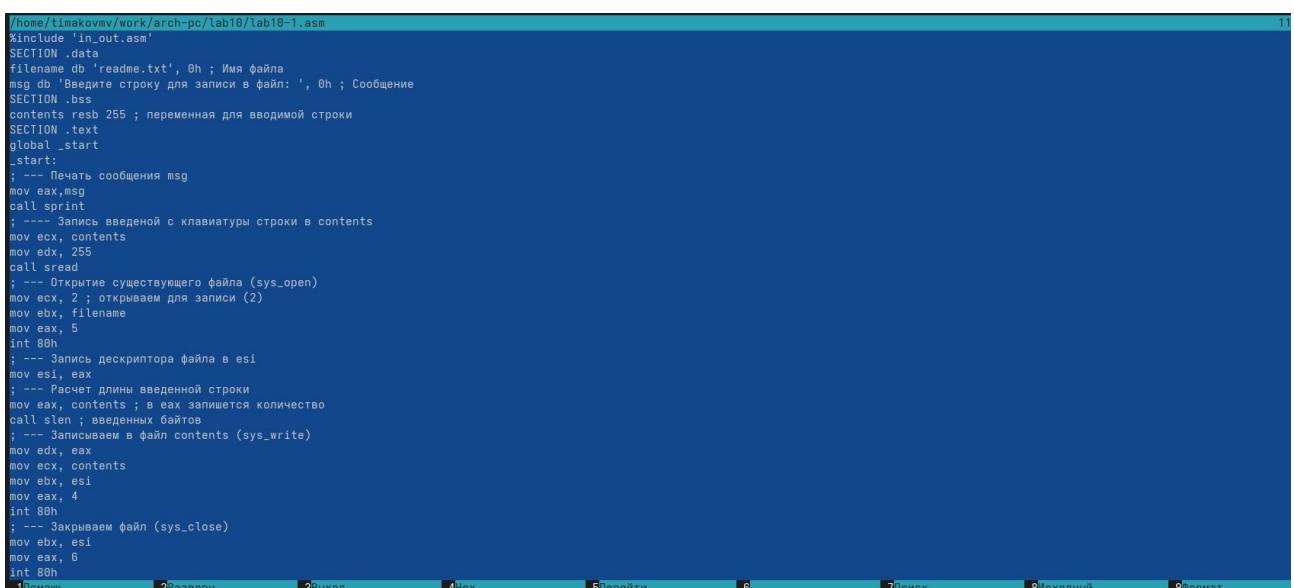
Создаем каталог для программ ЛБ10, и в нем создаем файлы (рис. Рисунок 2.1).

```
timakovmv@timakovmv:~/work/arch-pc/lab09$ cd ..
timakovmv@timakovmv:~/work/arch-pc$ mkdir lab10
timakovmv@timakovmv:~/work/arch-pc$ cd lab10
timakovmv@timakovmv:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt adme-2.txt
timakovmv@timakovmv:~/work/arch-pc/lab10$ █
```

Рисунок 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. Рисунок 2.2)



```
/home/timakovmv/work/arch-pc/lab10/lab10-1.asm
%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 ; 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
; --- 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
```

Рисунок 2.2: Заполняем файл

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

```
timakovmv@timakovmv:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
timakovmv@timakovmv:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
timakovmv@timakovmv:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello world!
timakovmv@timakovmv:~/work/arch-pc/lab10$
```

Рисунок 2.3: Запускаем файл и проверяем его работу

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

```
введите строку для записи в файл. нечто>отказ
timakovmv@timakovmv:~/work/arch-pc/lab10$ chmod -x lab10-1
timakovmv@timakovmv:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
timakovmv@timakovmv:~/work/arch-pc/lab10$
```

Рисунок 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. Рисунок 2.5).

```
bash: ./lab10-1: отказано в доступе
timakovmv@timakovmv:~/work/arch-pc/lab10$ chmod +x lab10-1.asm
timakovmv@timakovmv:~/work/arch-pc/lab10$ ./lab10-1.asm
./lab10-1.asm: строка 1: fg: нет управления заданиями
./lab10-1.asm: строка 2: SECTION: команда не найдена
./lab10-1.asm: строка 3: filename: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 7: SECTION: команда не найдена
./lab10-1.asm: строка 8: global: команда не найдена
./lab10-1.asm: строка 9: _start:: команда не найдена
./lab10-1.asm: строка 11: mov: команда не найдена
./lab10-1.asm: строка 12: call: команда не найдена
./lab10-1.asm: строка 13: mov: команда не найдена
./lab10-1.asm: строка 14: mov: команда не найдена
./lab10-1.asm: строка 15: call: команда не найдена
./lab10-1.asm: строка 16: mov: команда не найдена
./lab10-1.asm: строка 17: mov: команда не найдена
./lab10-1.asm: строка 18: mov: команда не найдена
./lab10-1.asm: строка 19: int: команда не найдена
./lab10-1.asm: строка 20: mov: команда не найдена
./lab10-1.asm: строка 21: mov: команда не найдена
./lab10-1.asm: строка 22: call: команда не найдена
./lab10-1.asm: строка 23: mov: команда не найдена
./lab10-1.asm: строка 24: mov: команда не найдена
./lab10-1.asm: строка 25: mov: команда не найдена
./lab10-1.asm: строка 26: mov: команда не найдена
./lab10-1.asm: строка 27: int: команда не найдена
./lab10-1.asm: строка 28: mov: команда не найдена
./lab10-1.asm: строка 29: mov: команда не найдена
```

Рисунок 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 19

Предоставляем права доступа к 2ум файлам, согласно варианту 19 в символьном и двоичном виде, затем проверяем работу команд. (рис. Рисунок 2.6).

```
timakovmv@timakovmv:~/work/arch-pc/lab10$ chmod u=rw,g=rwx,o=rx readme-1.txt
timakovmv@timakovmv:~/work/arch-pc/lab10$ chmod 505 readme-2.txt
timakovmv@timakovmv:~/work/arch-pc/lab10$ ls -l
итого 36
-rw-r--r--. 1 timakovmv timakovmv 3942 ноя 15 05:46 in_out.asm
-rw-r--r--. 1 timakovmv timakovmv 5640 ноя 25 18:26 lab10-1
-rwxr-xr-x. 1 timakovmv timakovmv 555 ноя 25 18:40 lab10-1.asm
-rw-r--r--. 1 timakovmv timakovmv 13518 ноя 25 18:26 lab10-1.lst
-rw-r--r--. 1 timakovmv timakovmv 2528 ноя 25 18:26 lab10-1.o
-rw-rwxr-x. 1 timakovmv timakovmv 0 ноя 25 18:15 readme-1.txt
-rx---r-x. 1 timakovmv timakovmv 0 ноя 25 19:40 readme-2.txt
timakovmv@timakovmv:~/work/arch-pc/lab10$ █
```

Рисунок 2.6: Используем команду chmod для установки нужных прав, после этого проверяем правильность выполнения командой ls -l

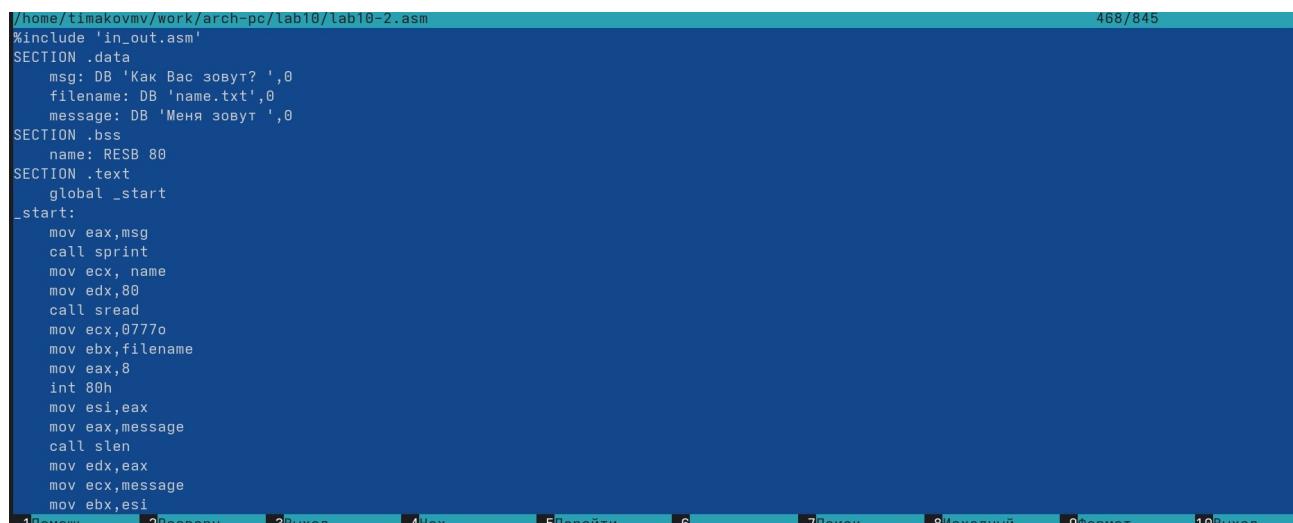
3. Задание для самостоятельной работы

Создаем новый файл (рис. Рисунок 2.7).

```
timakovmv@timakovmv:~/work/arch-pc/lab10$ touch lab10-2.asm  
timakovmv@timakovmv:~/work/arch-pc/lab10$
```

Рисунок 2.7: Создаем файл командой touch

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



```
/home/timakovmv/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 sprint  
    mov ecx, name  
    mov edx,80  
    call sread  
    mov ecx,07770  
    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
```

Рисунок 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. Рисунок 2.9).

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

Рисунок 2.9: Проверяем работу программы

4 Выводы

Мы научились писать программы для работы с файлам и научились предоставлять права доступа к файлам.