

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: Ибрагим Хиссеин Гана

Группа: НПИбд 01-25

МОСКВА

2025г.

Работа с файлами средствами NASM

Цель работы

Приобретение навыков написания программ для работы с файлами средствами NASM в ОС Linux, изучение системных вызовов для работы с файлами и механизмов управления правами доступа.

1. Подготовительный этап

1.1. Создание рабочего каталога и файлов

```
```bash
mkdir ~/work/arch-pc/lab10
cd ~/work/arch-pc/lab10
touch lab10-1.asm readme-1.txt readme-2.txt
```

```
итого 16
drwxrwxr-x 2 ibrahim ibrahim 4096 déc. 14 15:00 .
drwxrwxr-x 9 ibrahim ibrahim 4096 déc. 14 14:55 ..
-rw-rw-r-- 1 ibrahim ibrahim 3799 nov. 22 19:24 in_out.asm
-rw-rw-r-- 1 ibrahim ibrahim 562 déc. 14 14:56 lab10-1.asm
-rw-rw-r-- 1 ibrahim ibrahim 0 déc. 14 14:55 readme-1.txt
-rw-rw-r-- 1 ibrahim ibrahim 0 déc. 14 14:55 readme-2.txt
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ █
```

#### 1.2. Подготовка вспомогательного файла

Файл in\_out.asm был использован из предыдущих лабораторных работ, так как содержит необходимые функции для ввода/вывода (sprint, sread, slen, quit).

### 2. Основная программа lab10-1.asm

#### 2.1. Исходный код программы

Программа запрашивает у пользователя строку и записывает её в файл readme.txt.

```
%include 'in_out.asm'
SECTION .data
filename db 'readme.txt', 0h
msg db 'Введите строку для записи в файл: ', 0h
SECTION .bss
contents resb 255
SECTION .text
global _start
_start:
 mov eax, msg
 call sprint

 mov ecx, contents
 mov edx, 255
 call sread

 mov ecx, 2
 mov ebx, filename
 mov eax, 5
 int 80h

 mov esi, eax

 mov eax, contents
 call slen

 mov edx, eax
 mov ecx, contents
 mov ebx, esi
 mov eax, 4
 int 80h

 mov ebx, esi
 mov eax, 6
 int 80h

 call quit
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

## 2.2. Компиляция и сборка

nasm -f elf lab10-1.asm

ld -m elf\_i386 -o lab10-1 lab10-1.o

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ ls -la lab10-1.o
-rw-rw-r-- 1 ibrahim ibrahim 1472 déc. 14 15:04 lab10-1.o
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

```
-rwxrwxr-x 1 ibrahim ibrahim 9164 déc. 14 15:05 lab10-1
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

## 2.3. Тестирование программы

./lab10-1

При запросе программы была введена строка: Hello from NASM Lab 10!

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$./lab10-1
Введите строку для записи в файл: Hello from NASM Lab 10!
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

## 2.4. Проверка созданного файла

ls -l readme.txt cat readme.txt

cat readme.txt

Файл успешно создан и содержит введённый текст.

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ ls -l
-rw-r--r-- 1 ibrahim ibrahim 24 déc. 14 15:21 readme.txt
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

```
-rw-r--r-- 1 ibrahim ibrahim 24 déc. 14 15:25 readme.txt
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ cat readme.txt
Hello from NASM Lab 10!
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

### **3. Работа с правами доступа к файлам**

#### 3.1. Запрет выполнения файла lab10-1

```
chmod a-x lab10-1
```

```
ls -l lab10-1
```

```
./lab10-1
```

После снятия прав на выполнение, попытка запуска программы завершается ошибкой “Отказано в доступе”.

```
-rw-rw-r-- 1 ibrahim ibrahim 9256 déc. 14 15:23 lab10-1
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$./lab10-1
bash: ./lab10-1: Отказано в доступе
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

#### 3.2. Установка прав доступа по варианту 16

Вариант 16:

Символьный вид: -x r-x -w-

Двоичный вид: 001 010 101 (восьмеричный: 125)

```
chmod u=x,g=rx,o=w readme-1.txt
```

```
chmod 125 readme-2.txt
```

```
ls -l readme-*.txt
```

Права доступа успешно установлены для обоих файлов.

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ ls -l readme-*.txt
---x-r-x-w- 1 ibrahim ibrahim 0 déc. 14 14:55 readme-1.txt
---x-w-r-x 1 ibrahim ibrahim 0 déc. 14 14:55 readme-2.txt
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

#### 3.3. Проверка восьмеричных значений прав

```
stat -c "Fichier: %n | Octal: %a" readme-*.txt
```

Оба файла имеют восьмеричное значение прав 125, что соответствует варианту 16.

```
Fichier: readme-1.txt | Octal: 125
Fichier: readme-2.txt | Octal: 125
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

## 4. Самостоятельная работа

### 4.1. Программа name.asm

Программа запрашивает имя пользователя и записывает его в файл name.txt с префиксом **“Меня зовут”**.

```
%include 'in_out.asm'

SECTION .data
filename db 'name.txt', 0h
question db 'Как Вас зовут? ', 0h
prefix db 'Меня зовут ', 0h

SECTION .bss
name resb 255

SECTION .text
global _start
_start:
 mov eax, question
 call sprint

 mov ecx, name
 mov edx, 255
 call sread

 mov ecx, 06440
 mov ebx, filename
 mov eax, 8
 int 80h
 mov esi, eax

 mov eax, prefix
 call slen
 mov edx, eax
 mov ecx, prefix
 mov ebx, esi
 mov eax, 4
 int 80h

 mov eax, name
 call slen
 mov edx, eax
 mov ecx, name
 mov ebx, esi
 mov eax, 4
 int 80h

 mov ebx, esi
 mov eax, 6
```

### 4.2. Компиляция и тестирование

```
nasm -f elf name.asm
```

```
ld -m elf_i386 -o name name.o
```

```
./name
```

При запросе программы было введено имя: **Ибрахим**

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$./name
Как Вас зовут? Ибрахим
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$
```

### 4.3. Проверка результата

```
ls -l name.txt
```

```
cat name.txt
```

Файл успешно создан и содержит текст: **Меня зовут Ибрахим**

```
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ ls -l name.txt
-rw-r--r-- 1 ibrahim ibrahim 35 déc. 14 15:54 name.txt
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ cat name.txt
Меня зовут Ибрахим
ibrahim@ibrahim-IdeaPad-5-Pro-14ITL6:~/work/arch-pc/lab10$ █
```

## **5. Ответы на вопросы для самопроверки**

### **5.1. Как определяются права доступа к файлу в Unix-подобных ОС?**

Права доступа определяются для трёх категорий пользователей:

Владелец (user) — создатель файла

Группа (group) — пользователи, входящие в группу владельца

Остальные (others) — все остальные пользователи системы

Для каждой категории устанавливаются права на чтение (r), запись (w) и выполнение (x).

### **5.2. Как ОС определяет, является ли файл исполняемым?**

ОС определяет исполняемый файл по наличию права на выполнение (x) в правах доступа. Для запуска файла необходимо, чтобы у соответствующей категории пользователей было установлено право x.

### **5.3. Как разграничить права доступа для различных категорий пользователей?**

Для разграничения прав используются:

#### Символьный формат:

chmod u=rwx,g=rx,o=r файл

#### Восьмеричный формат:

chmod 754 файл (rwxr-xr-)

### **5.4. Номера системных вызовов для работы с файлами:**

sys\_read — 3

sys\_write — 4

sys\_open — 5

sys\_close — 6

sys\_creat — 8

sys\_lseek — 19

sys\_unlink — 10

### **5.5. Использование регистров в системных вызовах:**

EAX — номер системного вызова

EBX — первый аргумент (дескриптор файла или имя файла)

ECX — второй аргумент (адрес буфера или режим доступа)

EDX — третий аргумент (количество байтов или права доступа)

### 5.6. Что такое дескриптор файла?

**Дескриптор файла** — это уникальный числовой идентификатор (16-битное целое число), который операционная система присваивает открытому файлу. Дескриптор используется для всех последующих операций с файлом через системные вызовы.

## **6. Выводы**

В ходе выполнения лабораторной работы были приобретены практические навыки работы с файлами в ассемблере NASM:

Изучены системные вызовы Linux для работы с файлами: создание, открытие, чтение, запись и закрытие файлов.

Освоена работа с дескрипторами файлов — ключевыми идентификаторами для файловых операций.

Получен опыт компиляции и сборки NASM-программ под архитектуру i386.

Изучены механизмы управления правами доступа в Linux, включая символьную и восьмеричную нотацию.

Реализована автономная программа, взаимодействующая с пользователем и файловой системой.

Работа продемонстрировала эффективность использования системных вызовов для низкоуровневых операций с файлами и важность правильного управления правами доступа в многопользовательской системе.