

Отчёт по лабораторной работе №10

Дисциплина: архитектура компьютера

Федорова Анжелика Игоревна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Самостоятельная работа	14
6	Выводы	18
7	Список литературы	19

Список иллюстраций

4.1	Создание нужного каталога и файла lab10-1.asm	9
4.2	Копирование файла in-out.asm	9
4.3	Заполнение файла	10
4.4	Создание исполняемого файла	10
4.5	Проверка содержимого файла readme-1.txt	11
4.6	Изменение прав доступа с помощью chmod	11
4.7	Изменение прав доступа к файлу lab10-1.asm	12
4.8	Проверка прав на доступ	12
4.9	Предоставление прав на доступ по варианту 12	13
4.10	Предоставление прав на доступ для файла readme-2.txt	13
5.1	Копирование файла кода lab10-1.asm в name.asm	14
5.2	Редактирования файла name.asm	14
5.3	Исполнение файла name.asm	15
5.4	Проверка содержимого в name.txt	15

Список таблиц

1 Цель работы

Приобретение навыков написания программ для работы с файлами.

2 Задание

Напишите программу работающую по следующему алгоритму: • Вывод приглашения “Как Вас зовут?” • ввести с клавиатуры свои фамилию и имя • создать файл с именем name.txt • записать в файл сообщение “Меня зовут” • дописать в файл строку введенную с клавиатуры • закрыть файл

3 Теоретическое введение

С GNU/Linux является многопользовательской операционной системой. И для обеспечения защиты данных одного пользователя от действий других пользователей существуют специальные механизмы разграничения доступа к файлам. Кроме ограничения доступа, данный механизм позволяет разрешить другим пользователям доступ данным для совместной работы. Права доступа определяют набор действий (чтение, запись, выполнение), разрешённых для выполнения пользователям системы над файлами. Для каждого файла пользователь может входить в одну из трех групп: владелец, член группы владельца, все остальные. Для каждой из этих групп может быть установлен свой набор прав доступа. Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он имеет вид строк `gwx`, где вместо любого символа может стоять дефис. Всего возможно 8 комбинаций, приведенных в таблице 10.1. Буква означает наличие права (установлен в единицу второй бит триады `g` — чтение, первый бит `w` — запись, нулевой бит `x` — исполнение), а дефис означает отсутствие права (нулевое значение соответствующего бита). Также права доступа могут быть представлены как восьмеричное число. Так, права доступа `rw` (чтение и запись, без исполнения) понимаются как три двоичные цифры `110` или как восьмеричная цифра `6`.

Для создания и открытия файла служит системный вызов `sys_creat`, который использует следующие аргументы: права доступа к файлу в регистре `ECX`, имя файла в `EBX` и номер системного вызова `sys_creat` (8) в `EAX`.

Для записи в файл служит системный вызов `sys_write`, который использует

следующие аргументы: количество байтов для записи в регистре EDX, строку содержимого для записи ECX, файловый дескриптор в EBX и номер системного вызова `sys_write` (4) в EAX. Системный вызов возвращает фактическое количество записанных байтов в регистр EAX. В случае ошибки, код ошибки также будет находиться в регистре EAX. Прежде чем записывать в файл, его необходимо создать или открыть, что позволит получить дескриптор файла

Для чтения данных из файла служит системный вызов `sys_read`, который использует следующие аргументы: количество байтов для чтения в регистре EDX, адрес в памяти для записи прочитанных данных в ECX, файловый дескриптор в EBX и номер системного вызова `sys_read` (3) в EAX. Как и для записи, прежде чем читать из файла, его необходимо открыть, что позволит получить дескриптор файла.

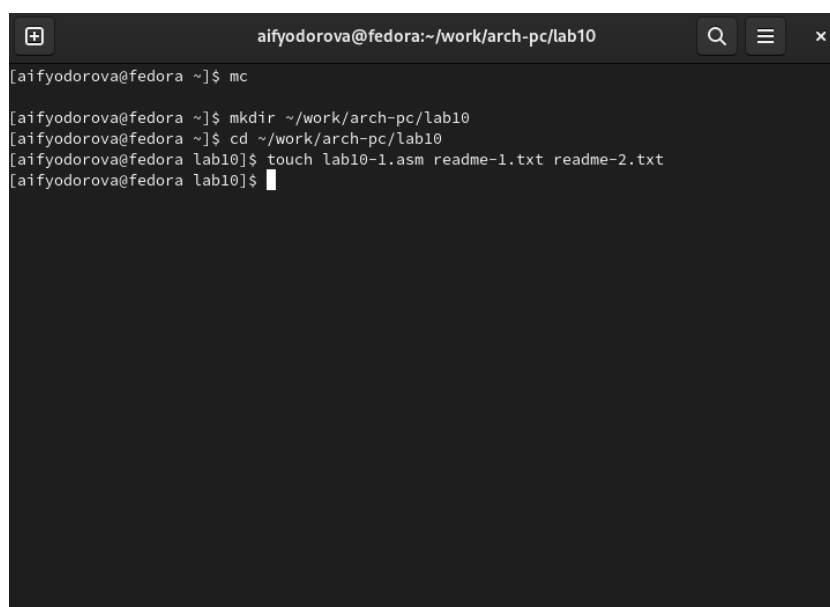
Для правильного закрытия файла служит системный вызов `sys_close`, который использует один аргумент – дескриптор файла в регистре EBX. После вызова ядра происходит удаление дескриптора файла, а в случае ошибки, системный вызов возвращает код ошибки в регистр EAX.

Для изменения содержимого файла служит системный вызов `sys_lseek`, который использует следующие аргументы: исходная позиция для смещения EDX, значение смещения в байтах в ECX, файловый дескриптор в EBX и номер системного вызова `sys_lseek` (19) в EAX. В случае ошибки, системный вызов возвращает код ошибки в регистр EAX.

Удаление файла осуществляется системным вызовом `sys_unlink`, который использует один аргумент – имя файла в регистре EBX.

4 Выполнение лабораторной работы

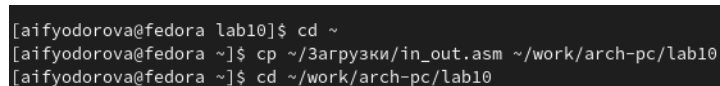
Создадим каталог для программ лабораторной работы и создадим в нем необходимые файлы (рис. fig:001).



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

Рис. 4.1: Создание нужного каталога и файла lab10-1.asm

Скопирую файл in-out.asm в нужный мне каталог. (рис. fig:002)



```
[aifyodorova@fedora lab10]$ cd ~
[aifyodorova@fedora ~]$ cp ~/Загрузки/in_out.asm ~/work/arch-pc/lab10
[aifyodorova@fedora ~]$ cd ~/work/arch-pc/lab10
```

Рис. 4.2: Копирование файла in-out.asm

Введём в файл lab10-1.asm текст программы из листинга (рис. fig:003).

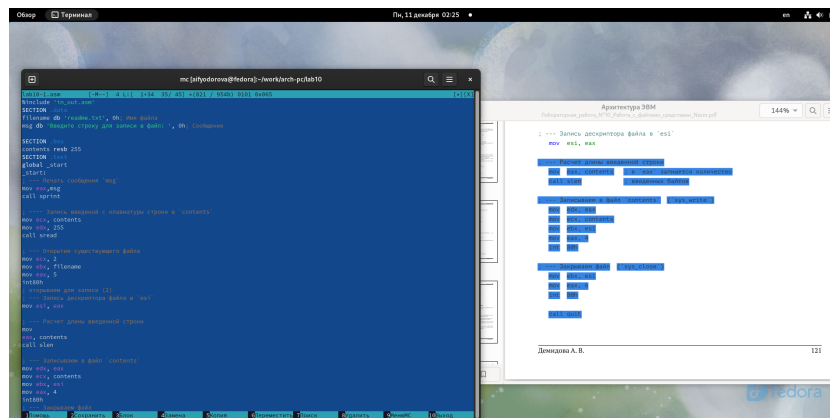


Рис. 4.3: Заполнение файла

Создадим исполняемый файл и проверим его работу (рис. fig:004).

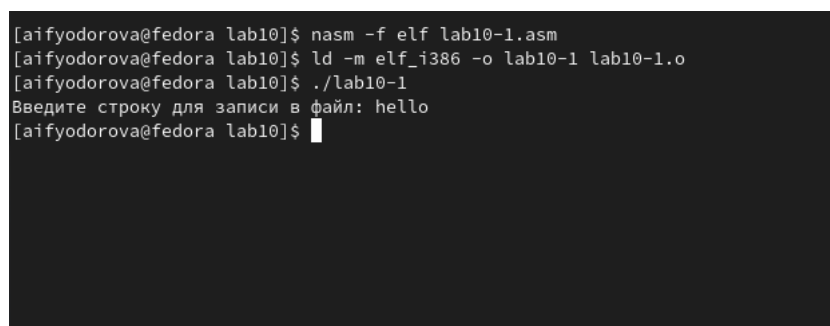


Рис. 4.4: Создание исполняемого файла

Теперь проверю содержимое файла readme-1.txt (рис.fig:005).

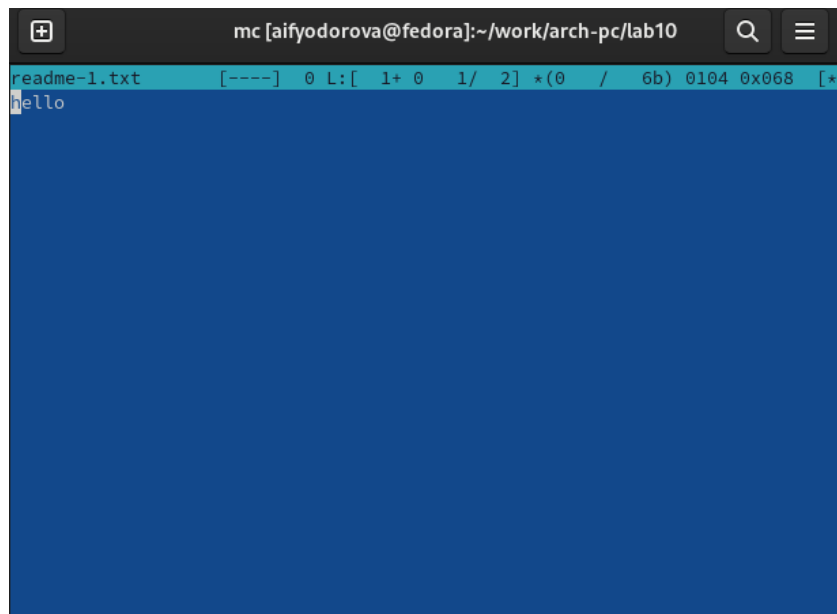


Рис. 4.5: Проверка содержимого файла readme-1.txt

Видно, что программа сработала успешно.

С помощью команды `chmod` изменим права доступа к исполняемому файлу `lab10-1`, запретив его выполнение (рис. fig:006).



Рис. 4.6: Изменение прав доступа с помощью `chmod`

Теперь доступ к файлу запрещён.

Изменим права доступа к файлу `lab10-1.asm`, добавив права на исполнение (рис. fig:007).

```
aifyodorova@fedora lab10$ chmod u+x lab10-1.asm
aifyodorova@fedora 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 3: Имя: команда не найдена
./lab10-1.asm: строка 4: msg: команда не найдена
./lab10-1.asm: строка 4: Сообщение: команда не найдена
./lab10-1.asm: строка 6: SECTION: команда не найдена
./lab10-1.asm: строка 7: contents: команда не найдена
./lab10-1.asm: строка 8: SECTION: команда не найдена
./lab10-1.asm: строка 9: global: команда не найдена
./lab10-1.asm: строка 10: _start: команда не найдена
./lab10-1.asm: строка 11: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 11: `'; --- Печать сообщения `msg'`
aifyodorova@fedora lab10$
```

Рис. 4.7: Изменение прав доступа к файлу lab10-1.asm

Файл нельзя выполнить, т.к. он не содержит указаний для терминала, а просто является текстом программы.

Проверю какие права имеются у пользователей файла readme-1.txt с помощью команды `ls` с ключом `-l`(рис. fig:008).

```
aifyodorova@fedora:~/work/arch-pc/lab10
[aifyodorova@fedora ~]$ mc
[aifyodorova@fedora lab10]$ ls -l readme-1.txt
-rw-r--r--. 1 aifyodorova aifyodorova 0 дек 11 02:13 readme-1.txt
[aifyodorova@fedora lab10]$
```

Рис. 4.8: Проверка прав на доступ

Предоставим права доступа к этому файлу в соответствии с вариантом 12 (рис.fig:009).

```
[aifyodorova@fedora lab10]$ chmod u+x readme-1.txt
[aifyodorova@fedora lab10]$ chmod u-rw readme-1.txt
[aifyodorova@fedora lab10]$ chmod go+x readme-1.txt
[aifyodorova@fedora lab10]$ chmod g-r ^C
[aifyodorova@fedora lab10]$ chmod g-r readme-1.txt
[aifyodorova@fedora lab10]$ chmod g+w readme-1.txt
[aifyodorova@fedora lab10]$ ls -l readme-1.txt
---x-wxr-x. 1 aifyodorova aifyodorova 0 дек 11 02:13 readme-1.txt
[aifyodorova@fedora lab10]$
```

Рис. 4.9: Предоставление прав на доступ по варианту 12

Прделаю то же самое, но с помощью двоичного кода с файлом readme-2.txt (рис.fig:010).

```
[aifyodorova@fedora lab10]$ chmod 122 readme-2.txt
[aifyodorova@fedora lab10]$ ls -l readme-2.txt
---x-w--w-. 1 aifyodorova aifyodorova 0 дек 11 02:13 readme-2.txt
[aifyodorova@fedora lab10]$
```

Рис. 4.10: Предоставление прав на доступ для файла readme-2.txt

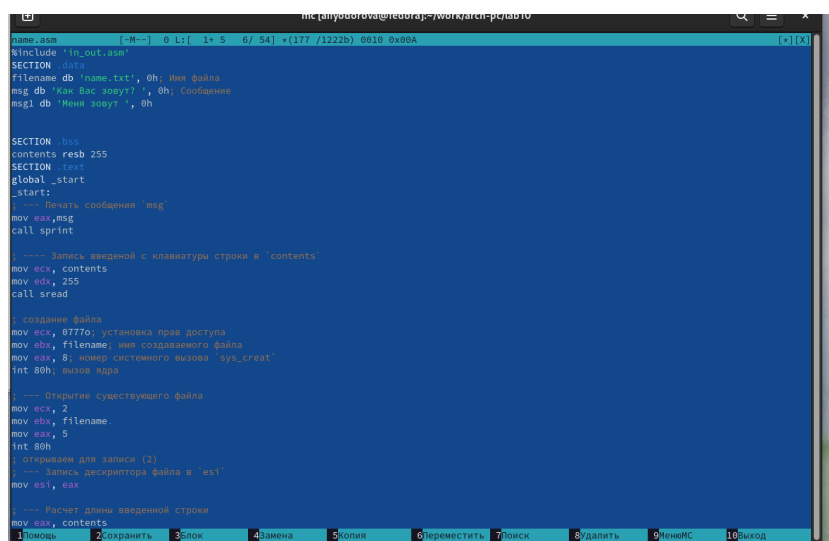
5 Самостоятельная работа

Скопирую содержимое файла lab10-1.asm в файла name.asm (рис.fig:011).

```
[aifyodorova@fedora lab10]$ cp lab10-1.asm name.asm
[aifyodorova@fedora lab10]$
```

Рис. 5.1: Копирование файла кода lab10-1.asm в name.asm

Редактирую код данного файла в соответствии с целью задания (рис.fig:012).



```
name.asm  [-*~]  0 L: 1+ 5  6/ 54) +(177 /1222b) 0010 0x00A  [+][X]
%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
SECTION .text
global _start
_start:
; --- Печата сообщение msg
mov eax, msg
call sprint

; --- Запись введенной с клавиатуры строки в 'contents'
mov ecx, contents
mov edi, 255
call read

; создание файла
mov ecx, 07770; установка прав доступа
mov ebx, filename; имя создаваемого файла
mov ebx, 8; номер системного вызова 'sys_create'
int 80h; вызов ядра

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

; открываем для записи (2)
; --- Записи дескриптора файла в 'esi'
mov esi, eax

; --- Расчет длины введенной строки
mov ebx, contents
1.Поместить 2.Сохранить 3.Блок 4.Замена 5.Скопия 6.Переместить 7.Поиск 8.Удалить 9.Вывод 10.Выход
```

Рис. 5.2: Редактирования файла name.asm

Создаю исполняемый файл и запускаю его. (рис.fig:013)

```
aifyodorova@fedora:~/work/arch-pc/lab10
[aifyodorova@fedora lab10]$ nasm -f elf name.asm
[aifyodorova@fedora lab10]$ ld -m elf_1386 -o name name.o
[aifyodorova@fedora lab10]$ ./name
Как Вас зовут? Анжелика
[aifyodorova@fedora lab10]$
```

Рис. 5.3: Исполнение файла name.asm

Теперь проверяю содержимое каталога с помощью ls (должен быть создан файл name.txt) и проверяю его содержимое с помощью команды cat(рис.fig:014)

```
[aifyodorova@fedora lab10]$ ls
in_out.asm lab10-1 lab10-1.asm lab10-1.o name name.asm name.o name.txt readme-1.txt readme-2.txt
[aifyodorova@fedora lab10]$ cat name.txt
Меня зовут Анжелика
[aifyodorova@fedora lab10]$
```

Рис. 5.4: Проверка содержимого в name.txt

Программа работает в соответствии с требованиями задания.

Код программы:

```
%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

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
mov ecx, 0777o; установка прав доступа
mov ebx, filename; имя создаваемого файла
mov eax, 8; номер системного вызова sys_creat
int 80h; вызов ядра

; --- Открытие существующего файла
mov ecx, 2
mov ebx, filename
mov eax, 5
int 80h
; открываем для записи (2)
; --- Запись дескриптора файла в esi
mov esi, eax

mov eax, msg1; в eax запишется количество
call slen ; введенных байтов

```



```

; --- Записываем в файл contents (sys_write)
mov edx, eax
mov ecx, msg1
mov ebx, esi
mov eax, 4
int 80h

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

; --- Записываем в файл contents

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

; --- Закрываем файл
mov ebx, esi
mov eax, 6
int 80h

call quit

```

6 Выводы

Я приобрела навыки написания программ для работы с файлами.

7 Список литературы

Лабораторная работа №10. Работа с файлами средствами Nasm