

Отчёт по лабораторной работе №10

Работа с файлами средствами Nasm.

Югай Александр Витальевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Задание для самостоятельной работы	10
3	Выводы	13

Список иллюстраций

2.1	Создаем каталог с помощью команды <code>mkdir</code> и файлы с помощью команды <code>touch</code>	6
2.2	Заполняем файл	7
2.3	Запускаем файл и проверяем его работу	8
2.4	Используем команду <code>chmod</code> для установки нужных прав, после этого пытаемся запустить файл	8
2.5	Используем команду <code>chmod</code> для установки нужных прав, после этого пытаемся запустить файл	9
2.6	Используем команду <code>chmod</code> для установки нужных прав, после этого проверяем правильность выполнения командой <code>ls -l</code>	9
2.7	Создаем файл командой <code>touch</code>	10
2.8	Пишем программу в <code>nano</code>	11
2.9	Проверяем работу программы	12

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

Создаем каталог для программ ЛБ10, и в нем создаем файлы

```
avyugayj@ubuntu:~$ mkdir ~/work/arch-pc/lab10
avyugayj@ubuntu:~$ cd ~/work/arch-pc/lab10/
avyugayj@ubuntu:~/work/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
avyugayj@ubuntu:~/work/arch-pc/lab10$
```

Рис. 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

```

GNU nano 6.2 /home
%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
call quit

```

Рис. 2.2: Заполняем файл

Создаем исполняемый файл и запускаем его

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

Рис. 2.3: Запускаем файл и проверяем его работу

Изменяем права доступа к файлу, запретив его выполнение. Пробуем запустить файл

```
avyugayj@ubuntu:~/work/arch-pc/lab10$ chmod -x lab10-1
avyugayj@ubuntu:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе
avyugayj@ubuntu:~/work/arch-pc/lab10$
```

Рис. 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


```

avyugayj@ubuntu:~/work/arch-pc/lab10$ chmod +x lab10-1.asm
avyugayj@ubuntu:~/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 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 5: SECTION: команда не найдена
./lab10-1.asm: строка 6: contents: команда не найдена
./lab10-1.asm: строка 6: переменная: 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 10: синтаксическая ошибка рядом с неожиданным маркером «;»
./lab10-1.asm: строка 10: `; --- Печать сообщения `msg`

```

Рис. 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-3

Предоставляем права доступа к 2ум файлам, согласно варианту 20 в символьном и двоичном виде, затем проверяем работу команд.

```

avyugayj@ubuntu:~/work/arch-pc/lab10$ chmod u=rx,g=wx,o=rw readme-1.txt
avyugayj@ubuntu:~/work/arch-pc/lab10$ chmod 353 readme-2.txt
avyugayj@ubuntu:~/work/arch-pc/lab10$ ls -l
итого 24
-rw-rw-r-- 1 avyugayj avyugayj 3942 ноя 11 00:28 in_out.asm
-rw-rw-r-- 1 avyugayj avyugayj 9164 дек 14 20:42 lab10-1
-rwxrwxr-x 1 avyugayj avyugayj 1140 дек 14 20:41 lab10-1.asm
-rw-rw-r-- 1 avyugayj avyugayj 1472 дек 14 20:42 lab10-1.o
-r-x-wxgw- 1 avyugayj avyugayj    0 дек 14 20:39 readme-1.txt
--wxr-x-wx 1 avyugayj avyugayj    0 дек 14 20:39 readme-2.txt
avyugayj@ubuntu:~/work/arch-pc/lab10$

```

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

2.1 Задание для самостоятельной работы

Создаем новый файл

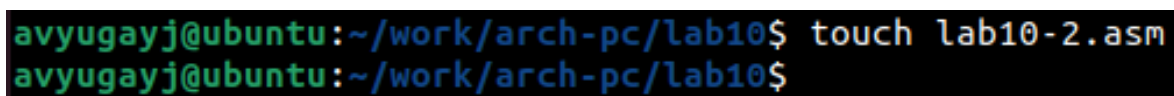
A terminal window with a black background and green text. The prompt is 'avyugayj@ubuntu:~/work/arch-pc/lab10\$'. The command 'touch lab10-2.asm' is entered and executed. The next line shows the prompt again: 'avyugayj@ubuntu:~/work/arch-pc/lab10\$'.

Рис. 2.7: Создаем файл командой touch

Пишем программу, которая выполнит представленный список действий

```

GNU nano 6.2
#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,0777o
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
mov eax,4
int 80h
mov ebx,esi
mov eax,6
int 80h
mov ecx,1
mov ebx,filename
mov eax,5
int 80h
mov esi,eax
mov edx,2
mov ecx,0
mov ebx,eax
mov eax,19
int 80h
mov eax,name

```

Рис. 2.8: Пишем программу в nano

Создаем исполняемый файл и запускаем его, после этого проверяем создался ли новый файл, затем смотрим, как он заполнен

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

Рис. 2.9: Проверяем работу программы

3 Выводы

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