

Отчёт по лабораторной работе №10

По теме: Работа с файлами средствами Nasm

Выполнил: Пателепень Филипп Максимович, НММбд-04-24.

10.1. Цель работы

Приобретение навыков написания программ для работы с файлами.

Выполнение лабораторной работы

10.3. Порядок выполнения лабораторной работы

1. Я создал каталог для программ лабораторной работы № 10, перешёл в него и создал файлы lab10-1.asm, readme-1.txt и readme-2.txt:

```
philipp916@philipp916-VirtualBox:~$ mkdir ~/work/arch-pc/lab10
philipp916@philipp916-VirtualBox:~$ cd ~/work/arch-pc/lab10
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ touch lab10-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ touch readme-1.txt readme-2.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$
```

2. Я ввел в файл lab10-1.asm текст программы из листинга 10.1 (Программа записи в файл сообщения). Создал исполняемый файл и проверил его работу. Ответ сохранился в файле readme-1.txt. С помощью команды cat я проверил правильность выполнения программы:

```

GNU nano 4.8 /home/philipp916/work/arch-pc/lab10/lab10-1.asm Изменён
%include 'in out.asm'
SECTION .data
filename db 'readme-1.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
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Hello World!
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ls -l
итого 48
-rw-rw-r-- 1 philipp916 philipp916 3942 ноя  9 17:03 in out.asm
-rwxrwxr-x 1 philipp916 philipp916 10948 дек 13 16:49 lab10-1
-rw-rw-r-- 1 philipp916 philipp916 472 дек 13 16:48 lab10-1.asm
-rw-rw-r-- 1 philipp916 philipp916 12618 дек 13 16:48 lab10-1.lst
-rw-rw-r-- 1 philipp916 philipp916 4272 дек 13 16:48 lab10-1.o
-rw-rw-r-- 1 philipp916 philipp916 13 дек 13 16:49 readme-1.txt
-rw-rw-r-- 1 philipp916 philipp916 0 дек 13 16:38 readme-2.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ cat readme-1.txt
Hello World!

```

- Далее с помощью команды `chmod` я изменил права доступа к исполняемому файлу `lab10-1`, запретив его выполнение. В терминале вывелся очевидный результат: отказ в доступе, ведь я командой `chmod` запретил запускать программу для владельца:

```

philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ chmod u-x lab10-1
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1
bash: ./lab10-1: Отказано в доступе

```

- Следующим шагом с помощью команды `chmod` я изменил права доступа к файлу `lab10-1.asm` с исходным текстом программы, добавив права на исполнение. Снова создал исполняемый файл и проверил его работу:

```

philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ chmod u+x lab10-1
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-1.lst lab10-1.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-1 lab10-1.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ./lab10-1
Введите строку для записи в файл: Philipp
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ cat readme-1.txt
Philipp
rld!

```

В результате программа заработала, но переписала исходный текст под новый, так как этот файл был со всеми разрешениями, и предыдущим шагом я запретил выполняться уже готовой программе. Система считает, что это новая программа, хотя она обладает другими разрешениями.

5. В соответствии с вариантом, полученным в ходе выполнения лабораторной работы №6, я предоставил права доступа к файлу readme1.txt представленные в символьном виде, а для файла readme-2.txt – в двочном виде. Проверил правильность выполнения с помощью команды ls -l:

```

philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ chmod a+x readme-1.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ls -l
итого 48
-rw-rw-r-- 1 philipp916 philipp916 3942 ноя  9 17:03 in_out.asm
-rwxrwxr-x 1 philipp916 philipp916 10948 дек 13 17:02 lab10-1
-rw-rw-r-- 1 philipp916 philipp916 472 дек 13 16:48 lab10-1.asm
-rw-rw-r-- 1 philipp916 philipp916 12618 дек 13 17:02 lab10-1.lst
-rw-rw-r-- 1 philipp916 philipp916 4272 дек 13 17:02 lab10-1.o
-rwxrwxr-x 1 philipp916 philipp916 13 дек 13 17:02 readme-1.txt
-rw-rw-r-- 1 philipp916 philipp916 0 дек 13 16:38 readme-2.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ chmod g-r+w readme-1.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ls -l
итого 48
-rw-rw-r-- 1 philipp916 philipp916 3942 ноя  9 17:03 in_out.asm
-rwxrwxr-x 1 philipp916 philipp916 10948 дек 13 17:02 lab10-1
-rw-rw-r-- 1 philipp916 philipp916 472 дек 13 16:48 lab10-1.asm
-rw-rw-r-- 1 philipp916 philipp916 12618 дек 13 17:02 lab10-1.lst
-rw-rw-r-- 1 philipp916 philipp916 4272 дек 13 17:02 lab10-1.o
-rwx-wxr-x 1 philipp916 philipp916 13 дек 13 17:02 readme-1.txt
-rw-rw-r-- 1 philipp916 philipp916 0 дек 13 16:38 readme-2.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ chmod 122 readme-2.txt
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ls -l
итого 48
-rw-rw-r-- 1 philipp916 philipp916 3942 ноя  9 17:03 in_out.asm
-rwxrwxr-x 1 philipp916 philipp916 10948 дек 13 17:02 lab10-1
-rw-rw-r-- 1 philipp916 philipp916 472 дек 13 16:48 lab10-1.asm
-rw-rw-r-- 1 philipp916 philipp916 12618 дек 13 17:02 lab10-1.lst
-rw-rw-r-- 1 philipp916 philipp916 4272 дек 13 17:02 lab10-1.o
-rwx-wxr-x 1 philipp916 philipp916 13 дек 13 17:02 readme-1.txt
---x-w--w- 1 philipp916 philipp916 0 дек 13 16:38 readme-2.txt

```

10.5. Задание для самостоятельной работы

1. Сначала я создал файл lab10-2.asm для выполнения самостоятельной работы:

```
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ touch lab10-test.asm
```

2. Далее я написал код для программы, которая будет запрашивать имя и выводить его в созданном файле name.txt, который создаст сама программа:

```
GNU nano 4.8 /home/philipp916/work/arch-pc/lab10/lab10-2.asm Изменён
#include "in_out.asm"

SECTION .data
file db 'name.txt', 0h
msg db 'Как вас зовут?', 0h
imya db 'Меня зовут:', 0
SECTION .bss
con resb 2550
SECTION .text
global _start
_start:

mov eax, msg
call sprintf
mov ecx, con
mov edx, 2550
call sread

mov ecx, 0777o
mov ebx, file
mov eax, 8
int 80h
mov esi, eax
mov eax, imya
call slen

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

mov edx, eax
mov ecx, con
mov ebx, esi
mov eax, 4
int 80h
mov ebx, esi
mov eax, 6
int 80h
call quit
```

3. Я создал исполняемый файл и запустил его. Программа ждала ввода моего имени в терминал, а затем создала файл, содержащий введенное с клавиатуры имя. С помощью команд cat и ls я проверил наличие файла, его содержимое и правильность выполнения своей программы:

```

philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ nasm -f elf -g -l lab10-2.lst lab10-2.asm
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ ./lab10-2
Как вас зовут?
Филипп
philipp916@philipp916-VirtualBox:~/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
philipp916@philipp916-VirtualBox:~/work/arch-pc/lab10$ cat name.txt
Меня зовут:Филипп

```

Выводы

В ходе выполнения данной лабораторной работы я приобрел полезные навыки написания программ для работы с файлами в ассемблере NASM.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. *Newham C.* Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. *Robbins A.* Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. *Zarrelli G.* Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. *Колдаев В. Д., Лупин С. А.* Архитектура ЭВМ. — М. : Форум, 2018.
10. *Куляс О. Л., Никитин К. А.* Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. *Новожилов О. П.* Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. *Робачевский А., Немнюгин С., Стесик О.* Операционная система UNIX. — 2-е изд. — БХВ-Петербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. *Столяров А.* Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.

15. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).

16. *Таненбаум Э., Бос Х.* Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).