

answer to labrotary work 10

Discipline: Computer Architecture

ali hosseinabadi

Content

1	Work Goal	5
2	Assignment	6
3	Theoretical Introduction	7
4	Performing the Laboratory Work	8
4.1	Independent Work Assignment	11
5	Conclusions	15
6	References	16

List of illustrations

4.1	Creating a working directory	8
4.2	Program of the first listing	8
4.3	Running the program of the first listing	9
4.4	Demonstration of the chmod command	9
4.5	Running the text file	10
4.6	Symbolic and numerical notations	10

List of Tables

1 Work Goal

Acquiring skills in writing programs for working with files.

2 Assignment

1. Creating files in programs.
2. Changing file permissions for different user groups.
3. Completing independent assignments based on the materials of the laboratory work.

3 Theoretical Introduction

The GNU/Linux OS is a multi-user operating system. To protect the data of one user from the actions of other users, special mechanisms for access control to files exist. Besides access restriction, this mechanism allows other users access to data for collaborative work.

4 Performing the Laboratory Work

I create a directory for the programs of laboratory work No. 10 (Fig. -fig. 4.1).

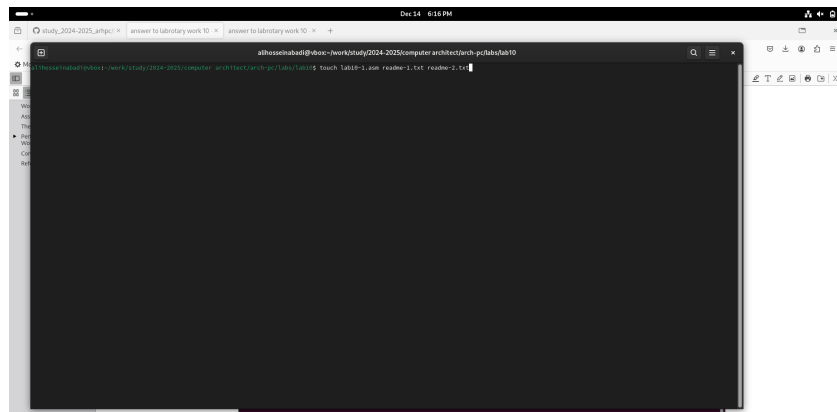


Fig. 4.1: Creating a working directory

I enter the program from the first listing into the created file (Fig. -fig. 4.2).

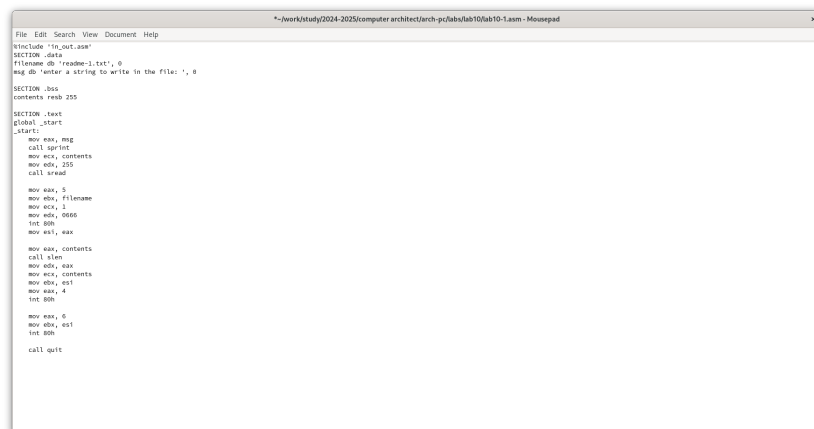
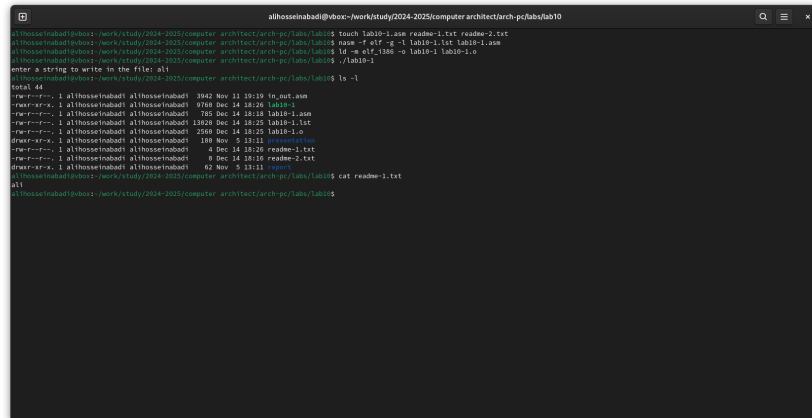


Fig. 4.2: Program of the first listing

I run the program; it prompts for a string input, after which it creates a text file with

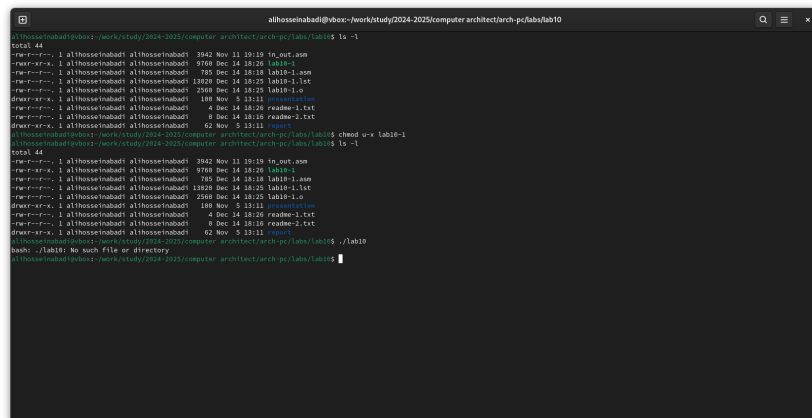
the string entered by the user (Fig. -fig. 4.3).



```
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ touch lab10-1.asm readme-1.txt readme-2.txt
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ nano -f ctf -g -l lab10-1.txt lab10-1.asm
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ ls -l && cat readme-1.txt
total 44
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 3942 Nov 11 19:19 in_out.asm
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 9768 Dec 14 18:26 lab10-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 785 Dec 14 18:18 lab10-1.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 13828 Dec 14 18:25 lab10-1.txt
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 2568 Dec 14 18:25 lab10-1.o
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 188 Nov 5 13:11 readme-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-1.txt
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 8 Dec 14 18:18 readme-2.txt
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 62 Nov 5 13:11 setup
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ cat readme-1.txt
all
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$
```

Fig. 4.3: Running the program of the first listing

I change the owner's permissions, prohibiting the execution of the file, after which the system refuses to execute the file because I, the owner, have prohibited myself from executing the program (Fig. -fig. 4.4).



```
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ ls -l
total 44
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 3942 Nov 11 19:19 in_out.asm
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 9768 Dec 14 18:26 lab10-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 785 Dec 14 18:18 lab10-1.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 13828 Dec 14 18:25 lab10-1.txt
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 2568 Dec 14 18:25 lab10-1.o
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 188 Nov 5 13:11 readme-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-1.txt
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 8 Dec 14 18:18 readme-2.txt
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 62 Nov 5 13:11 setup
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ chmod u-x lab10-1
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ ls -l
total 44
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 3942 Nov 11 19:19 in_out.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 9768 Dec 14 18:26 lab10-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 785 Dec 14 18:18 lab10-1.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 13828 Dec 14 18:25 lab10-1.txt
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 2568 Dec 14 18:25 lab10-1.o
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 188 Nov 5 13:11 readme-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-1.txt
-rwxr-xr-x 1 alhosseinabadi alhosseinabadi 8 Dec 14 18:18 readme-2.txt
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 62 Nov 5 13:11 setup
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$ cd /lab10
alhosseinabadi@vbox: /lab10$ ls
alhosseinabadi@vbox: /lab10$ No such file or directory
alhosseinabadi@vbox: /work/study/2024-2025/computer-architecture/arch-pc/lab10$
```

Fig. 4.4: Demonstration of the chmod command

I add the execution permission to the owner for the source program file; the executable text file interprets each line as a command. Since none of the lines are bash commands, the program does absolutely nothing (Fig. -fig. 4.5).

```
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10
total 44
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 3942 Nov 11 19:19 to_out.asm
-rw-r-xr-x 1 alhosseinabadi alhosseinabadi 9768 Dec 14 18:26 lab10-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 785 Dec 14 18:18 lab10-1.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 13828 Dec 14 18:25 lab10-1.txt
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 2568 Dec 14 18:25 lab10-1.o
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 168 Nov 5 13:11 .
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-1.txt
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-2.txt
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 62 Nov 5 13:11 .
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ chmod u-x lab10-1
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ ls -l
total 44
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 3942 Nov 11 19:19 to_out.asm
-rw-r-xr-x 1 alhosseinabadi alhosseinabadi 9768 Dec 14 18:26 lab10-1
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 785 Dec 14 18:18 lab10-1.asm
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 13828 Dec 14 18:25 lab10-1.txt
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 2568 Dec 14 18:25 lab10-1.o
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 168 Nov 5 13:11 .
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:26 readme-1.txt
-rw-r--r-- 1 alhosseinabadi alhosseinabadi 4 Dec 14 18:18 readme-2.txt
drwxr-xr-x 1 alhosseinabadi alhosseinabadi 62 Nov 5 13:11 .
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$
bash: ./lab10: No such file or directory
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ chmod u-x lab10-1
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ ./lab10-1.asm
bash: ./lab10-1.asm: Permission denied
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$
```

Fig. 4.5: Running the text file

According to my variant, I need to set the corresponding permissions to the text files created at the beginning of the laboratory work:

1. In symbolic form for the 1st readme file `-x -w- -w-`
2. In binary system for the 2nd readme file `001 011 101`

I convert the group of bits to the octal system; I adjust the symbolic notation to the syntax and obtain the necessary arguments for `chmod` (Fig. -fig. 4.6).

```
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ chmod u-x,g-x,o-x readme-1.txt
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ chmod 1551 readme-2.txt
bash: chmod: command not found...
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$ chmod 1551 readme-2.txt
alhosseinabadi@vbox: ~/work/study/2024-2025/computer_architect/arch-pc/labs/lab10$
```

Fig. 4.6: Symbolic and numerical notations

4.1 Independent Work Assignment

I write a program, transliterate and compile it. The program should display a prompt, request input from the keyboard, and create a text file with the string specified in the program and the user's input.

I run the program, check the presence and content of the created text file; the program

```
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ touch lab10-2.asm
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ mousepad lab10-2.asm
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ nano -f elf lab10-2.asm
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ ld -m elf_i386 -o lab10-2 lab10-2.o
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ ./lab10-2
what is your name?hi
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ ls
in.out.asm lab10-1.asm lab10-1.txt lab10-1.o lab10-2 lab10-2.asm lab10-2.o name.txt presentation reader-1.txt reader-2.txt report
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$ cat name.txt
my name ali
alhosseinabadi@vbox:~/work/study/2024-2025/computer-architect/arch-pc/labs/lab10$
```

works correctly (Fig. -fig. ??).

```

File Edit Search View Document Help
~/haskell/2024-2025/computer-architecture/ch04-pcSubLab10.hs:10:2: Misparsed

#include "is_not.asm"

SECTION .data
    filename db "name.txt", 0
    prompt db "What is your name?", 0
    letter db "Bye name ", 0

SECTION .bss
    name resb 255

SECTION .text
    global _start
    _start:
        mov esi, prompt
        call %print

        mov esi, name
        mov ebx, 255
        call %read

        mov eax, 0
        mov ebx, %ebxname
        mov esi, %ebx
        int $0x

    mov esi, eax
        int $0x

        mov eax, %ebx
        call %write

        mov ebx, esi
        mov esi, %ebxname
        mov ebx, %ebx
        int $0x

        mov esi, 0
        int $0x

        mov esi, name
        call %write

        mov ebx, esi
        mov esi, %ebxname
        mov ebx, %ebx
        int $0x

        mov ebx, esi
        int $0x

        mov ebx, 0
        int $0x

        call %quit

```

Program code:

```
%include 'in_out.asm'
```

SECTION .data

```
filename db 'name.txt', 0
```

```
prompt db 'what is your name?', 0
```

```
intro db 'my name is ', 0
```

```
SECTION .bss
```

```
name resb 255
```

```
SECTION .text
```

```
global _start
```

```
_start:
```

```
mov eax, prompt
```

```
call sprint
```

```
mov ecx, name
```

```
mov edx, 255
```

```
call sread
```

```
mov eax, 8
```

```
mov ebx, filename
```

```
mov ecx, 0744o
```

```
int 80h

mov esi, eax

mov eax, intro

call slen

mov edx, eax

mov ecx, intro

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
```

```
int 80h
```

```
call quit
```

5 Conclusions

In the process of performing the laboratory work, I acquired skills in writing programs for working with files and learned how to edit file permissions.

6 References

1. Course on TUIS
2. Programming in NASM Assembler Language Stolyarov A. V.