

Отчет по лабораторной работе №5

Архитектура компьютера

Садыков Ильдар Ильфатович

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	7
3.1 Работа с Midnight Commander:	7
3.2 Редактирование программы lab5-1.asm:	8
3.3 Трансляция и запуск программы:	9
3.4 Работа с внешним файлом in_out.asm:	10
4 Задания для самостоятельной работы	12
4.1 Программа без использования in_out.asm:	12
4.2 Программа с использованием in_out.asm:	13
5 Выводы	15

Список иллюстраций

3.1	Запуск Midnight Commander и переход в рабочий каталог	7
3.2	Редактирование файла lab5-1.asm в mcedit	8
3.3	Проверяем файл через F3 в Midnight Commander	9
3.4	Трансляция, компоновка и запуск программы lab5-1	10
3.5	Запуск программы lab5-2	11
4.1	Запуск и код программы lab5-1.asm	13
4.2	Запуск и код программы lab5-2.asm	14

Список таблиц

1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

2 Задание

1. Освоить основы работы с файловым менеджером Midnight Commander.
2. Изучить структуру программы на языке ассемблера NASM.
3. Освоить использование системных вызовов в ОС GNU Linux.
4. Написать и запустить программу вывода сообщения на экран и ввода строки с клавиатуры.
5. Научиться подключать внешние файлы и использовать подпрограммы.

3 Выполнение лабораторной работы

3.1 Работа с Midnight Commander:

Запускаем Midnight Commander и переходим в рабочий каталог (рис. 3.1).

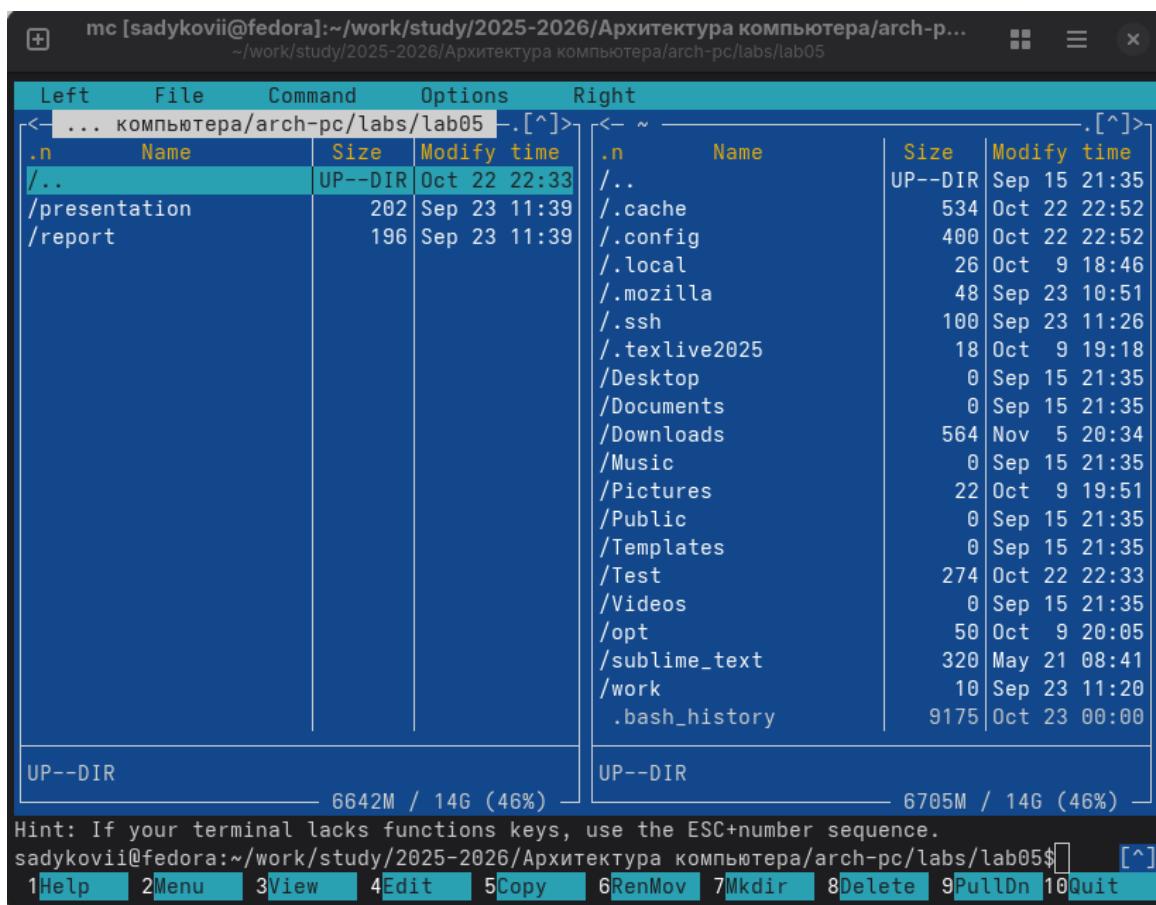
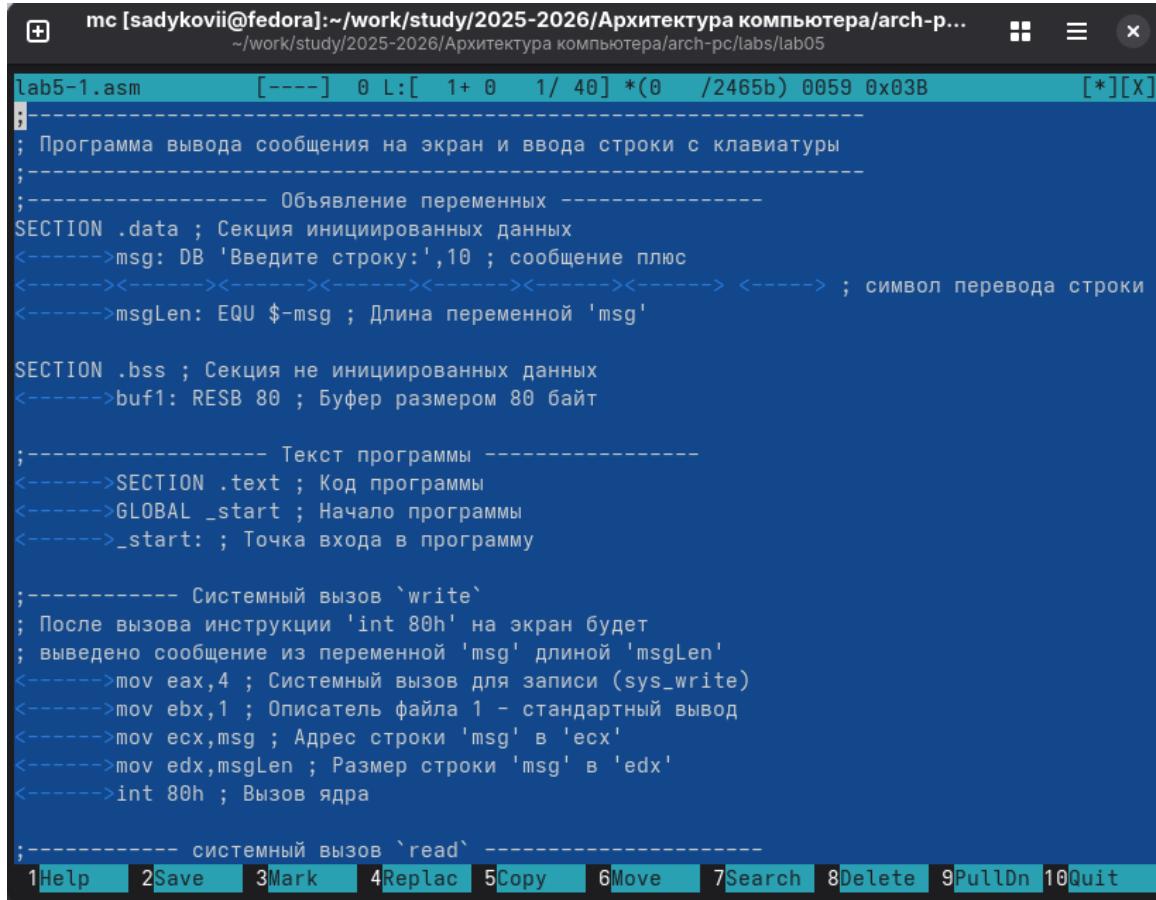


Рисунок 3.1: Запуск Midnight Commander и переход в рабочий каталог

3.2 Редактирование программы lab5-1.asm:

Создаем файл lab5-1.asm с помощью команды touch. Открываем файл lab5-1.asm во встроенным редакторе mcedit с помощью клавиши F4 и вводим код программы (рис. 3.2).



The screenshot shows the mcedit editor window with the file 'lab5-1.asm' open. The code is as follows:

```
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;----- Объявление переменных -----
SECTION .data ; Секция инициализированных данных
<---->msg: DB 'Введите строку:',10 ; сообщение плюс
<----><----><----><----><----><----><----> ; символ перевода строки
<---->msgLen: EQU $-msg ; Длина переменной 'msg'

SECTION .bss ; Секция не инициализированных данных
<---->buf1: RESB 80 ; Буфер размером 80 байт

;----- Текст программы -----
<---->SECTION .text ; Код программы
<---->GLOBAL _start ; Начало программы
<---->_start: ; Точка входа в программу

;----- Системный вызов `write`
; После вызова инструкции 'int 80h' на экран будет
; выведено сообщение из переменной 'msg' длиной 'msgLen'
<---->mov eax,4 ; Системный вызов для записи (sys_write)
<---->mov ebx,1 ; Описатель файла 1 - стандартный вывод
<---->mov ecx,msg ; Адрес строки 'msg' в 'ecx'
<---->mov edx,msgLen ; Размер строки 'msg' в 'edx'
<---->int 80h ; Вызов ядра

;----- системный вызов `read` -----
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Рисунок 3.2: Редактирование файла lab5-1.asm в mcedit

Убеждаемся в правильности написания кода([рис. @fig-003]).

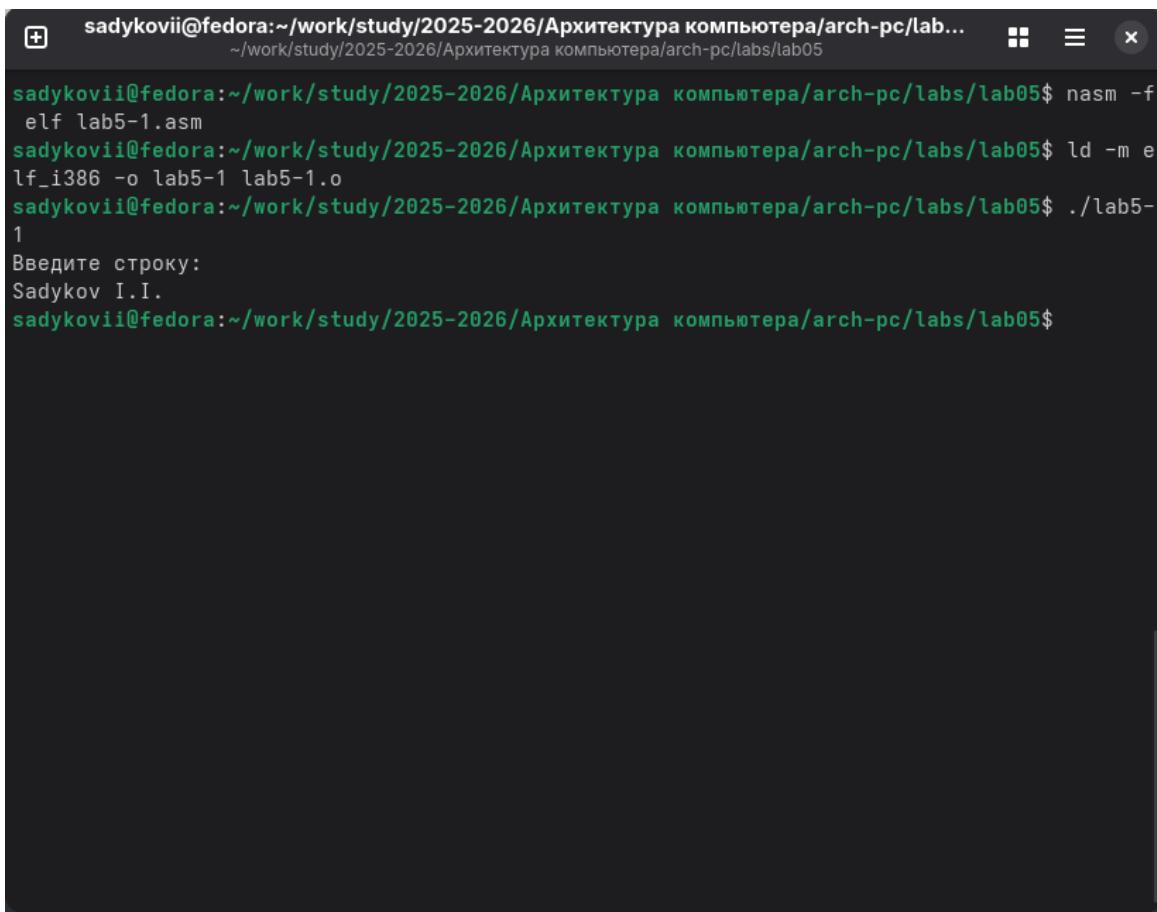
The screenshot shows a terminal window titled 'mc [sadykovii@fedora]:~/work/study/2025-2026/Архитектура компьютера/arch-p...' displaying an assembly code file. The file contains comments and assembly instructions for a program that reads input from the keyboard and writes it to the screen. The assembly code includes sections for data (.data), bss (.bss), and text (.text). It defines variables like msg and msgLen, and uses system calls like int 80h for write and read operations. The bottom of the window shows a menu bar with options 1 through 10.

```
;-----  
; Программа вывода сообщения на экран и ввода строки с клавиатуры  
;  
;----- Объявление переменных -----  
SECTION .data ; Секция инициализированных данных  
    msg: DB 'Введите строку:',10 ; сообщение плюс  
                                ; символ перевода строки  
    msgLen: EQU $-msg ; Длина переменной 'msg'  
  
SECTION .bss ; Секция не инициализированных данных  
    buf1: RESB 80 ; Буфер размером 80 байт  
  
;----- Текст программы -----  
SECTION .text ; Код программы  
GLOBAL _start ; Начало программы  
_start: ; Точка входа в программу  
  
;----- Системный вызов `write`  
; После вызова инструкции 'int 80h' на экран будет  
; выведено сообщение из переменной 'msg' длиной 'msgLen'  
    mov eax,4 ; Системный вызов для записи (sys_write)  
    mov ebx,1 ; Описатель файла 1 - стандартный вывод  
    mov ecx,msg ; Адрес строки 'msg' в 'ecx'  
    mov edx,msgLen ; Размер строки 'msg' в 'edx'  
    int 80h ; Вызов ядра  
  
;----- системный вызов `read` -----  
1Help 2UnWrap 3Quit 4Hex 5Goto 6 7Search 8Raw 9Format 10Quit
```

Рисунок 3.3: Проверяем файл через F3 в Midnight Commander

3.3 Трансляция и запуск программы:

Выполняем трансляцию и компоновку программы, затем запускаем исполняемый файл (рис. 3.4).



```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f elf lab5-1.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-1
Введите строку:
Sadykov I.I.
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$
```

Рисунок 3.4: Трансляция, компоновка и запуск программы lab5-1

3.4 Работа с внешним файлом in_out.asm:

Копируем файл in_out.asm в текущий каталог. Создаем копию файла lab5-1.asm с именем lab5-2.asm с помощью клавиши F6. Редактируем lab5-2.asm с использованием подпрограмм из in_out.asm и запускаем файл(рис. 3.5).

The screenshot shows a terminal window with the following session:

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ mc
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ ^C
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ mc
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ mc
[1]+  Stopped                  /usr/bin/mc -P "$MC_PWD_FILE" "$@"
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ mc
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ nasm -f
elf lab5-2.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ ld -m e
lf_i386 -o lab5-2 lab5-2.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$ ./lab5-
2
Введите строку:
Sadykov I.I
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05$
```

Рисунок 3.5: Запуск программы lab5-2

4 Задания для самостоятельной работы

4.1 Программа без использования in_out.asm:

Модифицируем lab5-1.asm для ввода и вывода строки (рис. 4.1).

Left	File	Command	Options	Right	
<code>... kompyutera/arch-pc/labs/lab05 .[^\n]</code>				<code>~ .[^\n]</code>	
<code>.n Name</code>	<code>Size</code>	<code>Modify time</code>	<code>.n Name</code>	<code>Size</code>	<code>Modify time</code>
<code>/.. UP--DIR</code>		<code>Oct 22 22:33</code>	<code>..</code>	<code>UP--DIR</code>	<code>Sep 15 21:35</code>
<code>/presentation 202</code>		<code>Sep 23 11:39</code>	<code>./cache 534</code>		<code>Oct 22 22:52</code>
<code>/report 196</code>		<code>Sep 23 11:39</code>	<code>./config 400</code>		<code>Oct 22 22:52</code>
			<code>./local 26</code>		<code>Oct 9 18:46</code>
			<code>./mozilla 48</code>		<code>Sep 23 10:51</code>
			<code>./ssh 100</code>		<code>Sep 23 11:26</code>
			<code>./texlive2025 18</code>		<code>Oct 9 19:18</code>
			<code>/Desktop 0</code>		<code>Sep 15 21:35</code>
			<code>/Documents 0</code>		<code>Sep 15 21:35</code>
			<code>/Downloads 564</code>		<code>Nov 5 20:34</code>
			<code>/Music 0</code>		<code>Sep 15 21:35</code>
			<code>/Pictures 22</code>		<code>Oct 9 19:51</code>
			<code>/Public 0</code>		<code>Sep 15 21:35</code>
			<code>/Templates 0</code>		<code>Sep 15 21:35</code>
			<code>/Test 274</code>		<code>Oct 22 22:33</code>
			<code>/Videos 0</code>		<code>Sep 15 21:35</code>
			<code>/opt 50</code>		<code>Oct 9 20:05</code>
			<code>/sublime_text 320</code>		<code>May 21 08:41</code>
			<code>/work 10</code>		<code>Sep 23 11:20</code>
			<code>.bash_history 9175</code>		<code>Oct 23 00:00</code>
			<code>UP--DIR</code>	<code>6705M / 14G (46%)</code>	
			<code>UP--DIR</code>	<code>6642M / 14G (46%)</code>	
<p>Hint: If your terminal lacks functions keys, use the ESC+number sequence.</p> <p>sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs/lab05\$ [^]</p>					
<p>1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit</p>					

Рисунок 4.1: Запуск и код программы lab5-1.asm

4.2 Программа с использованием `in_out.asm`:

Модифицируем lab5-2.asm для ввода и вывода строки (рис. 4.2).

The screenshot shows a terminal window with the following session:

```
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$ nasm -f elf lab5-2.asm
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$ ld -m elf_i386 -o lab5-2 lab5-2.o
sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$ ./lab5-2
Введите строку:
Sadykov 1/1/
Sadykov 1/1

sadykovii@fedora:~/work/study/2025-2026/Архитектура компьютера/arch-pc/labs$
```

Below the terminal is a Sublime Text window showing the assembly code for `lab5-2.asm`:

```
7 SECTION .bss ; Секция не инициализированных данных
8     buf1: RESB 80 ; Буфер размером 80 байт
9
10 ;----- Текст программы -----
11 SECTION .text ; Код программы
12     GLOBAL _start ; Начало программы
13     _start: ; Точка входа в программу
14
15     mov eax,msg
16     call sprintLF
17
18     mov ecx,buf1
19     mov edx,80
20
21     call sread
22 ;-----Повтор вводимой строки-----
23     mov eax,buf1
24     call sprintLF
25
26     call quit
```

Рисунок 4.2: Запуск и код программы lab5-2.asm

5 Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки работы с файловым менеджером Midnight Commander. Была изучена структура программ на языке ассемблера NASM и освоены инструкции mov и int. Написаны и запущены программы для вывода сообщений на экран и ввода строк с клавиатуры. Освоена работа с внешними файлами и подпрограммами, что позволяет упростить написание и чтение ассемблерных программ.