

Отчёт по лабораторной работе №4

дисциплина: Архитектура компьютера

Аносов Даниил Игоревич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Задание для самостоятельной работы	18
4.1	Первая часть задания	18
4.2	Вторая часть задания	19
5	Выводы	22

Список иллюстраций

3.1	Установка mc	7
3.2	Открытый mc	7
3.3	Переход в каталог курса	8
3.4	Создание директории в Midnight Commander	9
3.5	Переход в созданный каталог	10
3.6	Создание .asm-файла в новом каталоге	11
3.7	Открытый файл в редакторе внутри Midnight Commander	12
3.8	Файл с кодом в Midnight Commander	13
3.9	Компиляция и запуск программы	13
3.10	Открытый Midnight Commander с каталогом Downloads	14
3.11	Копирование скачанного файла в рабочий каталог	14
3.12	Копирование первого файла с кодом	15
3.13	Редактирование нового файла	16
3.14	Компиляция и запуск новой программы	16
3.15	Изменение команды в новой программе	17
3.16	Запуск программы с новой командой	17
4.1	Копирование файла lab5-1.asm	18
4.2	Редактирование файла lab5-1-copy.asm	19
4.3	Компиляция и вызов программы lab5-1-copy.asm	19
4.4	Редактирование файла lab5-2-copy.asm	20
4.5	Компиляция и вызов программы lab5-2-copy.asm	20
4.6	Загрузка файлов на GitHub	21

Список таблиц

1 Цель работы

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

2 Задание

1. Создайте копию файла lab5-1.asm. Внесите изменения в программу (без использования внешнего файла in_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.
3. Создайте копию файла lab5-2.asm. Исправьте текст программы с использование подпрограмм из внешнего файла in_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.

3 Выполнение лабораторной работы

Откроем терминал и установим пакет **mc** - *Midnight Commander* с помощью пакетного менеджера **pacman**. (рис. 3.1).

```
[dianoov@archlinux report]$ sudo pacman -S mc
(sudo) password for dianoov:
resolving dependencies...
looking for conflicting packages...

Packages (1) mc-4.8.32-1
Total Download Size: 1.81 MiB
Total Installed Size: 7.05 MiB

:: Proceed with installation? [Y/n]
:: Retrieving packages...
mc-4.8.32-1-x86_64 1851.0 KiB 2.64 MiB/s 00:01 [#####] 100%
(I/!) checking keys in keyring [#####] 100%
(I/!) checking package integrity [#####] 100%
(I/!) loading package files [#####] 100%
(I/!) checking for file conflicts [#####] 100%
(I/!) checking available disk space [#####] 100%
:: Processing package changes...
(I/!) installing mc
optional dependencies for mc
aspell: spelling corrections
camextract: oemv extra
cdparanoia: audio extra (installed)
cdrtit: iso9660 extra
cval: OIS support
gawk: hp4x extra (installed)
mcutils: ar extra
P7zip: support for 7zip archives
perl: needed by several extra scripts (installed)
python: to access util4j or sj storage (installed)
unace: uace extra
unrar: unrar extra
unzip: open zip archives (installed)
zip: unzip extra
:: Running post-transaction hooks...
(I/!) Arming ConditionNeedsUpdate...
[dianoov@archlinux report]$ mc
```

Рис. 3.1: Установка mc

Введём команду *mc*, чтобы открыть Midnight Commander. (рис. 3.2).

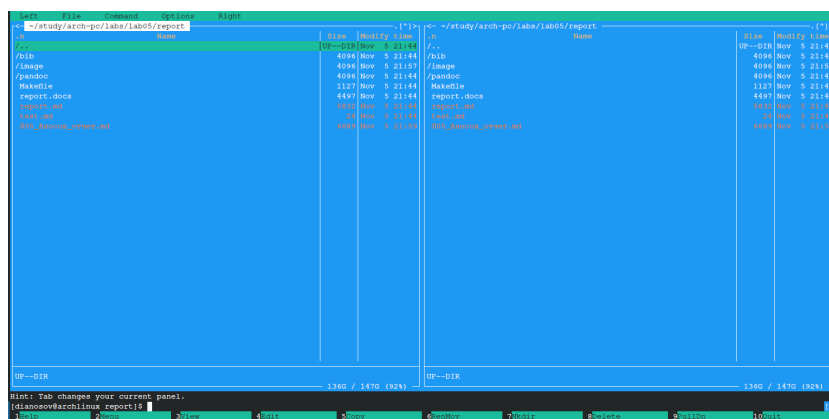


Рис. 3.2: Открытый mc

В Midnight Commander в одной из панелей откроем каталог курса *~/study/arch-pc*

(рис. 3.3).

```
Left      File      Command      Options      Right
< ~ /study/arch-pc .[ ^ > < ...dy/arch-pc/labs/lab05/report -. [ ^ >
.n      Name      Size      Modify time      .n      Name      Size      Modify time
UP--DIR      UP--DIR      UP--DIR      UP--DIR
./      ..      4096      Oct 26 13:14      ./      ..      4096      Nov 5 21:44
/.git      4096      Oct 26 23:33      /.bib      4096      Nov 5 21:44
/config      4096      Oct 26 13:16      /image      4096      Nov 5 21:57
/labs      4096      Nov 5 21:44      /pandoc      4096      Nov 5 21:44
/presentation      4096      Oct 26 13:16      Makefile      1127      Nov 5 21:44
/template      4096      Oct 26 13:16      report.docs      4497      Nov 5 21:44
.gitattributes      1765      Oct 26 13:16      report.md      5832      Nov 5 21:44
.gitignore      4637      Oct 26 13:16      test.md      24      Nov 5 21:44
.gitmodules      278      Oct 26 13:16      J05_Anocob_o949p.md      6689      Nov 5 21:59
CHANGELOG.md      4786      Oct 26 13:16
COURSE      8      Oct 26 13:16
HI_QiYsKI-Xxw.woff2      1864      Oct 26 13:16
LICENSE      18657      Oct 26 13:16
Makefile      980      Oct 26 13:16
README.en.md      152      Oct 26 13:16
README.git-flow.md      5653      Oct 26 13:16
README.md      4336      Oct 26 13:16
README.pdf      15313      Oct 26 13:16
README.tex      4454      Oct 26 13:16
prepare      0      Oct 26 13:16
*update.sh      53      Oct 26 23:27

*update.sh      136G / 147G (92%)      UP--DIR      136G / 147G (92%)

Hint: Setting the CDPATH variable can save you keystrokes in cd commands.
[dianosov@archlinux arch-pc]$
1Help      2Menu      3View      4Edit      5Copy      6RenMov      7Mkdir      8Delete      9FullDn      10Quit
```

Рис. 3.3: Переход в каталог курса

С помощью функциональной клавиши F7 создадим папку lab05. (рис. 3.4).

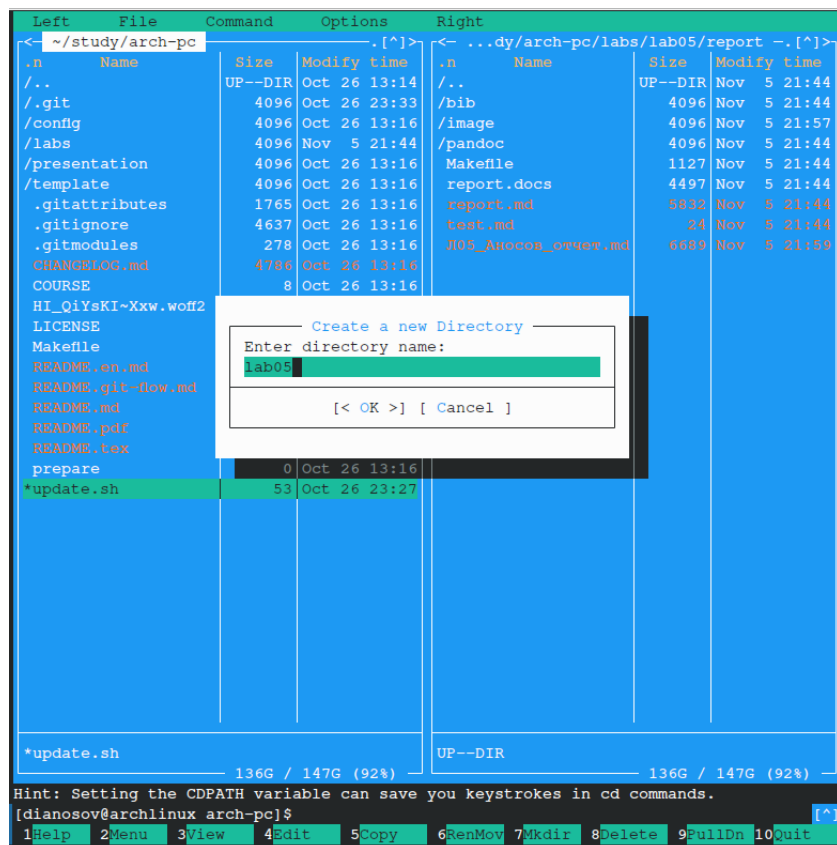


Рис. 3.4: Создание директории в Midnight Commander

Перейдём в созданный каталог. (рис. 3.5).

Left	File	Command	Options	Right
~ / study / arch-pc / lab05 . [^] >				< - ... dy / arch-pc / labs / lab05 / report - . [^] >
.n	Name	Size	Modify time	.n Name Size Modify time
/..		UP--DIR	Nov 5 23:57	/.. UP--DIR Nov 5 21:44
lab5-1.asm			Nov 5 23:58	/bib 4096 Nov 5 21:44
				/image 4096 Nov 5 23:58
				/pandoc 4096 Nov 5 21:44
				Makefile 1127 Nov 5 21:44
				report.docs 4497 Nov 5 21:44
				report.md 5832 Nov 5 21:44
				test.md 24 Nov 5 21:44
				009_Autocon_ot4et.md 5886 Nov 5 23:58
UP--DIR				UP--DIR
136G / 147G (92%)				136G / 147G (92%)
Hint: M-! will allow you to execute programs and see the output in the viewer.				
[dianosov@archlinux lab05]\$ touch lab5-1.asm				
1 Help 2 Menu 3 View 4 Edit 5 Copy 6 RenMov 7 Mkdir 8 Delete 9 FullDn 10 Quit				

Рис. 3.6: Создание .asm-файла в новом каталоге

С помощью функциональной клавиши F4 откроем файл lab5-1.asm для редактирования во встроенном редакторе. (рис. 3.7).

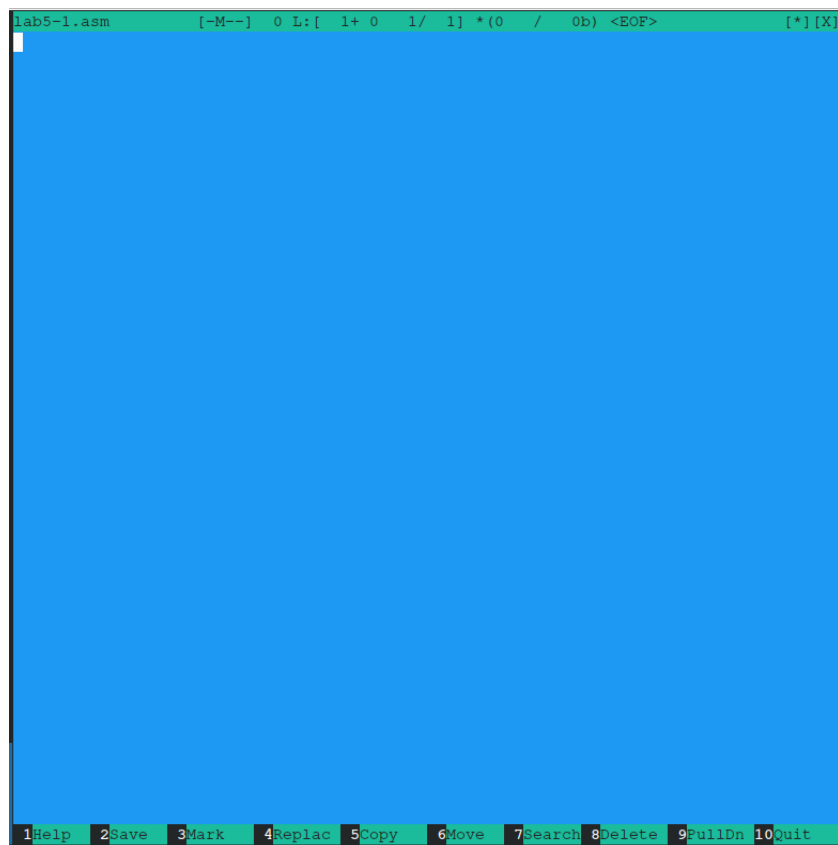


Рис. 3.7: Открытый файл в редакторе внутри Midnight Commander

```

lab5-1.asm      [-M--] 38 L:[ 1+ 5    6/ 35] *(465 /2431b) 1080 0x438      [*] [X]
;-----
; 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 ; Описание 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' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра

```

Рис. 3.8: Файл с кодом в Midnight Commander

Скомпилируем и запустим созданную программу lab5-1.asm (рис. 3.9).

```

[dianosov@archlinux arch-pc]$ cd lab05/
[dianosov@archlinux lab05]$ ls
lab5-1.asm
[dianosov@archlinux lab05]$ pwd
/home/dianosov/study/arch-pc/lab05
[dianosov@archlinux lab05]$ nasm -f elf lab5-1.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[dianosov@archlinux lab05]$ ./lab5-1
Введите строку:
Аносов Даниил Игоревич
[dianosov@archlinux lab05]$

```

Рис. 3.9: Компиляция и запуск программы

В одной из панелей mc откроем каталог с файлом lab5-1.asm. В другой панели каталог со скачанным файлом in_out.asm (рис. 3.10).



Рис. 3.10: Открытый Midnight Commander с каталогом Downloads

Скачаем и скопируем в текущую директорию файл in_out.asm с сайта ТУИС.
(рис. 3.11).

```
[dianosov@archlinux lab05]$ nasm -f elf lab5-1.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-1 lab5-1.o
[dianosov@archlinux lab05]$ ./lab5-1
Введите строку:
Аносов Даниил Игоревич
[dianosov@archlinux lab05]$ cp ~/Downloads/in_out.asm .
[dianosov@archlinux lab05]$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o
[dianosov@archlinux lab05]$
```

Рис. 3.11: Копирование скачанного файла в рабочий каталог

Скопируем первый файл с кодом и дадим новому файлу имя lab5-2.asm (рис. 3.12).

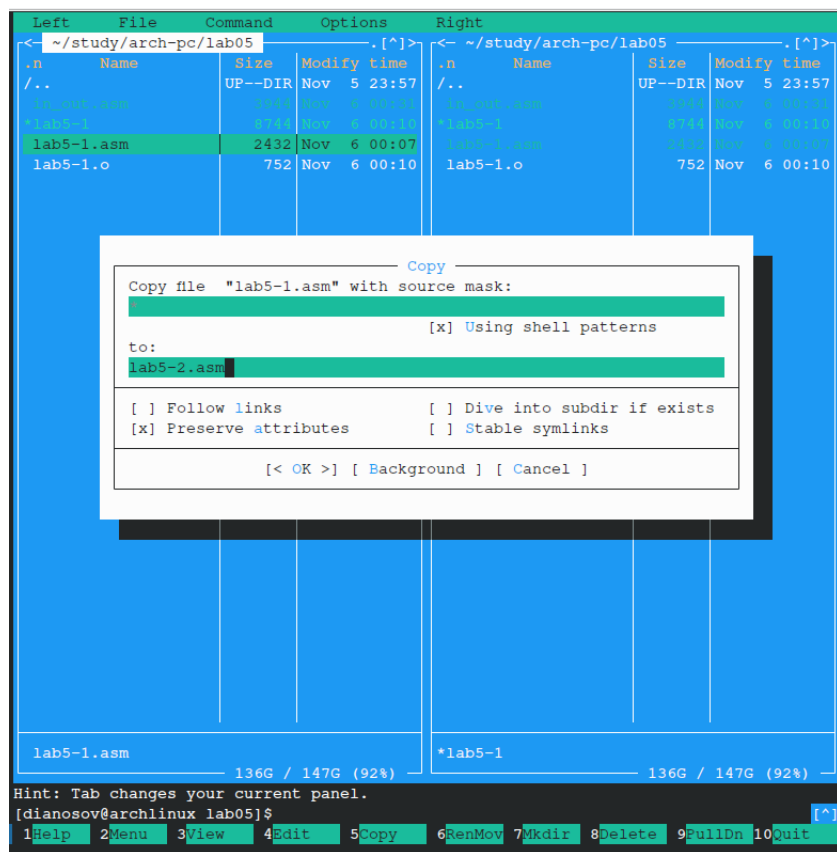


Рис. 3.12: Копирование первого файла с кодом

Отредактируем новый файл в соответствии с предложенным листингом. (рис. 3.13).

```

lab5-2.asm      [----]  0 L:[ 1+ 0  1/ 17] *(0  /1224b) 0059 0x03B  [*] [X]
;
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;
#include "in_out.asm" ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprintf ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины выводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.13: Редактирование нового файла

Проведем компиляцию нового отредактированного файла и проверим его работу. (рис. 3.14).

```

[dianosov@archlinux lab05]$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2.asm
[dianosov@archlinux lab05]$ nasm -f elf lab5-2.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[dianosov@archlinux lab05]$ ./lab5-2
Введите строку:
Аносов Даниил Игоревич
[dianosov@archlinux lab05]$

```

Рис. 3.14: Компиляция и запуск новой программы

В новой программе поменяем команду **sprintf** на **sprint**. (рис. 3.15).


```

lab5-2.asm      [BM--] 11 L:[ 1+12 13/ 17] *(847 /1222b) 0032 0x020  [*][X]
;~~~~~
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;~~~~~
#include "in_out.asm" ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB "Введите строку: ",0h ; сообщение
SECTION .bss ; Секция не инициализированных данных
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
mov eax, msg ; запись адреса выводимого сообщения в 'EAX'
call sprintLF ; вызов подпрограммы печати сообщения
mov ecx, buf1 ; запись адреса переменной в 'EAX'
mov edx, 80 ; запись длины вводимого сообщения в 'EBX'
call sread ; вызов подпрограммы ввода сообщения
call quit ; вызов подпрограммы завершения

```

Рис. 3.15: Изменение команды в новой программе

Проведем компиляцию нового отредактированного файла и проверим его работу, проанализировав, что изменилось. (рис. 3.16).

```

[dianosov@archlinux lab05]$ nasm -f elf lab5-2.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-2 lab5-2.o
[dianosov@archlinux lab05]$ ./lab5-2
Введите строку: Аносов Даниил Игоревич
[dianosov@archlinux lab05]$

```

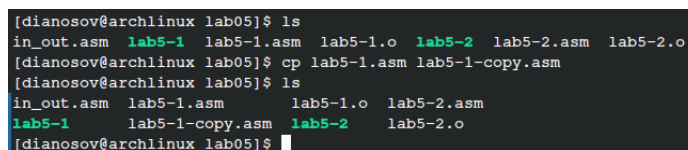
Рис. 3.16: Запуск программы с новой командой

Как видно, команда **sprintLF** переводит строку после двоеточия, а команда **sprint** оставляет курсор на той же строке.

4 Задание для самостоятельной работы

4.1 Первая часть задания

Скопируем файл *lab5-1.asm* для дальнейшего редактирования. (рис. 4.1).



```
[dianosov@archlinux lab05]$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.o
[dianosov@archlinux lab05]$ cp lab5-1.asm lab5-1-copy.asm
[dianosov@archlinux lab05]$ ls
in_out.asm lab5-1.asm lab5-1.o lab5-2.asm
lab5-1 lab5-1-copy.asm lab5-2 lab5-2.o
[dianosov@archlinux lab05]$
```

Рис. 4.1: Копирование файла lab5-1.asm

В коде добавим системный вызов *sys_write*, чтобы не только принимать строку от пользователя, но и выводить её после этого, как требуется в задании (код прокомментирован). (рис. 4.2).

```

;----- 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' -----
; После вызова инструкции 'int 80h' программа будет ожидать ввода
; строки, которая будет записана в переменную 'buf1' размером 80 байт
mov eax,3 ; Системный вызов для чтения (sys_read)
mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
mov ecx,buf1 ; Адрес буфера под вводимую строку
mov edx,80 ; Длина вводимой строки
int 80h ; Вызов ядра

mov eax,4 ; Системный вызов sys_write
mov ebx,1 ; Дескриптор stdout (терминал)
; в ecx уже находится адрес введенной строки
; в edx уже указана максимальная длина строки - 80
int 80h ; Вызов ядра
;----- Системный вызов 'exit' -----
; После вызова инструкции 'int 80h' программа завершит работу
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
~
~
-- INSERT --

```

Рис. 4.2: Редактирование файла lab5-1-copy.asm

Скомпилируем новую версию программы и проверим работоспособность. Как видно, введенная строка вывелась после того, как была введена в программу пользователем. Задание выполнено корректно. (рис. 4.3).

```

[dianosov@archlinux lab05]$ nasm -f elf lab5-1-copy.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-1-copy lab5-1-copy.o
[dianosov@archlinux lab05]$ ./lab5-1-copy
Введите строку:
Аносов Даниил Игоревич
Аносов Даниил Игоревич
[dianosov@archlinux lab05]$

```

Рис. 4.3: Компиляция и вызов программы lab5-1-copy.asm

4.2 Вторая часть задания

Скопируем файл *lab5-2.asm* для дальнейшего редактирования аналогично первой части задания, откроем его в редакторе *vim*. (рис. 4.4). Требуется сделать

так, чтобы программа выводила введённую пользователем строку, как и в первом задании. Но нужно использовать подключенный файл *in_out.asm*

[illegible]

Рис. 4.4: Редактирование файла lab5-2-сору.asm

Скомпилируем новую версию программы и проверим работоспособность. Как видно, введенная строка вывелась после того, как была введена в программу пользователем. Задание выполнено корректно. (рис. 4.5). Отметим, что, поскольку была использована подпрограмма **sprintLF** из файла *in_out.asm*, после выведенной строки также появилась пустая строка. Это было оговорено ранее в отчёте.

```
[dianosov@archlinux lab05]$ nasm -f elf lab5-2-copy.asm
[dianosov@archlinux lab05]$ ld -m elf_i386 -o lab5-2-copy lab5-2-copy.o
[dianosov@archlinux lab05]$ ./lab5-2-copy
Введите строку: Аносов Даниил Игоревич
Аносов Даниил Игоревич

[dianosov@archlinux lab05]$
```

Рис. 4.5: Компиляция и вызов программы lab5-2-sory.asm

Все задания выполнены, загрузим же новую версию проекта курса на Github.
(рис. 4.6).

```
[dianosov@archlinux arch-pc]$ git add .
[dianosov@archlinux arch-pc]$ git commit -am "add files for lab05"
On branch master
nothing to commit, working tree clean
[dianosov@archlinux arch-pc]$ git push origin master
Enumerating objects: 55, done.
Counting objects: 100% (55/55), done.
Delta compression using up to 4 threads
Compressing objects: 100% (51/51), done.
Writing objects: 100% (51/51), 4.40 MiB | 5.79 MiB/s, done.
Total 51 (delta 12), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (12/12), completed with 3 local objects.
To https://github.com/exterminateddd/pc-course-2024-2025
   bf17549..d351d6f  master -> master
[dianosov@archlinux arch-pc]$
```

Рис. 4.6: Загрузка файлов на GitHub

5 Выводы

В ходе выполнения данной лабораторной работы были приобретены практические навыки работы в Midnight Commander. Освоены инструкции языка ассемблера `mov` и `int`.