

Лабораторная работа №6

Дисциплина: Архитектура компьютера

Апареев Дмитрий Андреевич

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Задания для самостоятельной работы | 13 |
| 5 | Вывод | 17 |

Список иллюстраций

| | | | |
|-----|----|-----------|----|
| 3.1 | 1 | | 7 |
| 3.2 | 3 | | 8 |
| 3.3 | 4 | | 9 |
| 3.4 | 5 | | 9 |
| 3.5 | 6 | | 10 |
| 3.6 | 7 | | 10 |
| 3.7 | 8 | | 11 |
| 3.8 | 10 | | 11 |
| 3.9 | 11 | | 12 |
| 4.1 | 12 | | 13 |
| 4.2 | 13 | | 14 |
| 4.3 | 14 | | 14 |
| 4.4 | 15 | | 15 |
| 4.5 | 16 | | 15 |
| 4.6 | 17 | | 16 |

Список таблиц

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

1. Основы работы с тс
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

Открываю Midnight Commander, введя в терминал mc (рис. 3.1).

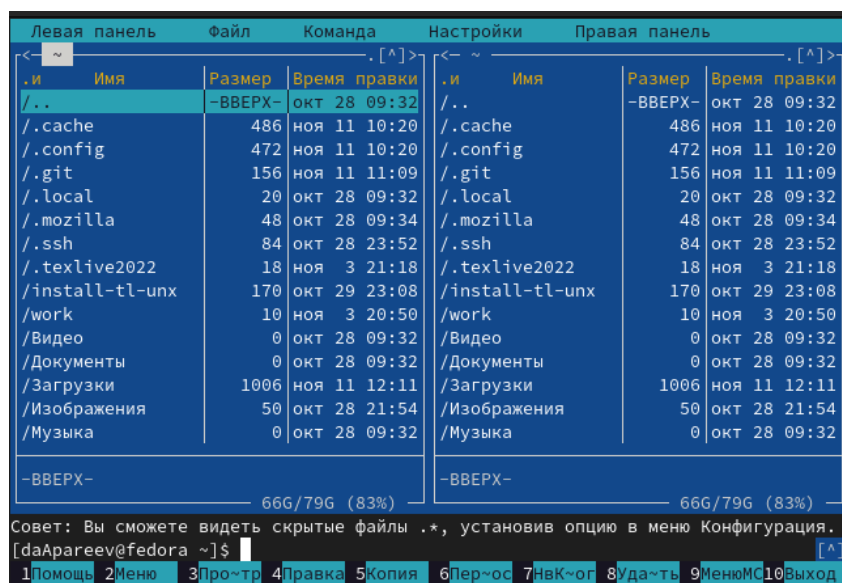
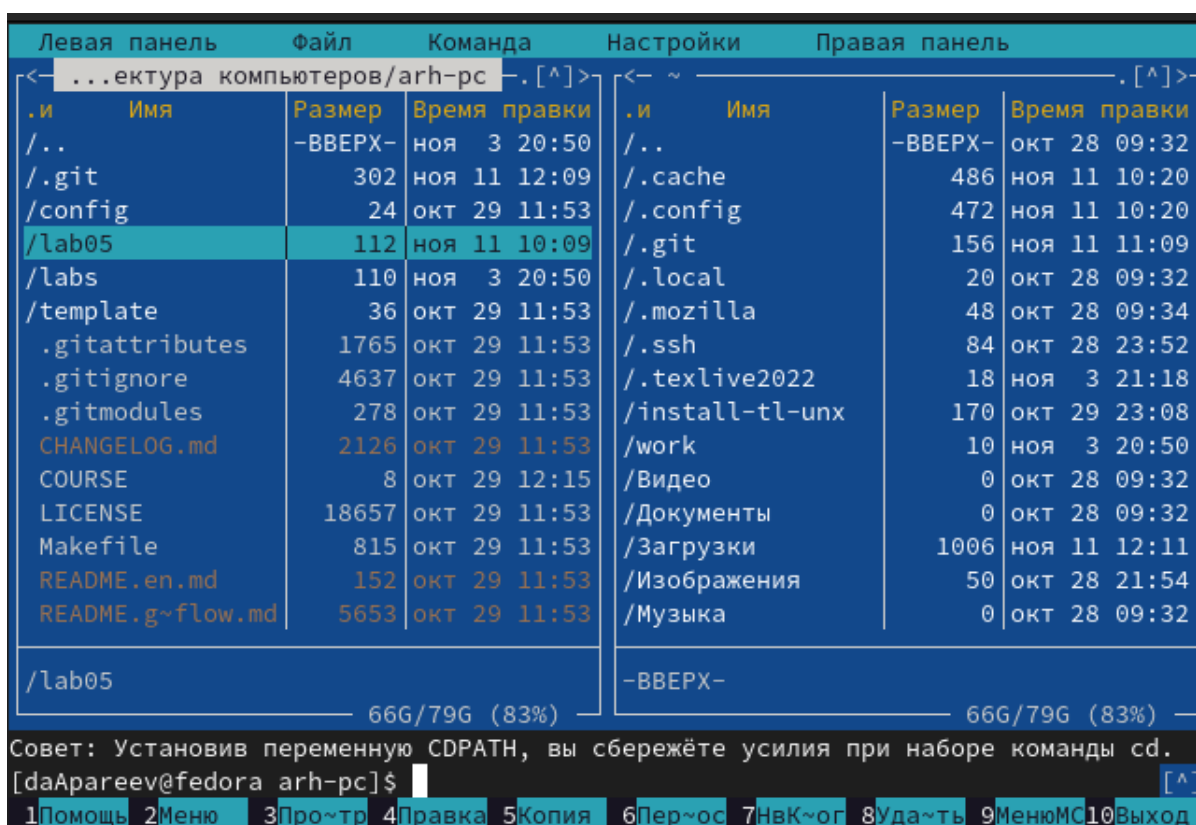


Рис. 3.1: 1

Перехожу в каталог ~/work/study/2022-2023/Архитектура Компьютера/arch-рс, используя файловый менеджер mc (рис. ??)



{ #fig:002 width=70% }

С помощью функциональной клавиши F7 создаю каталог lab06 и перехожу в него(рис. 3.2).

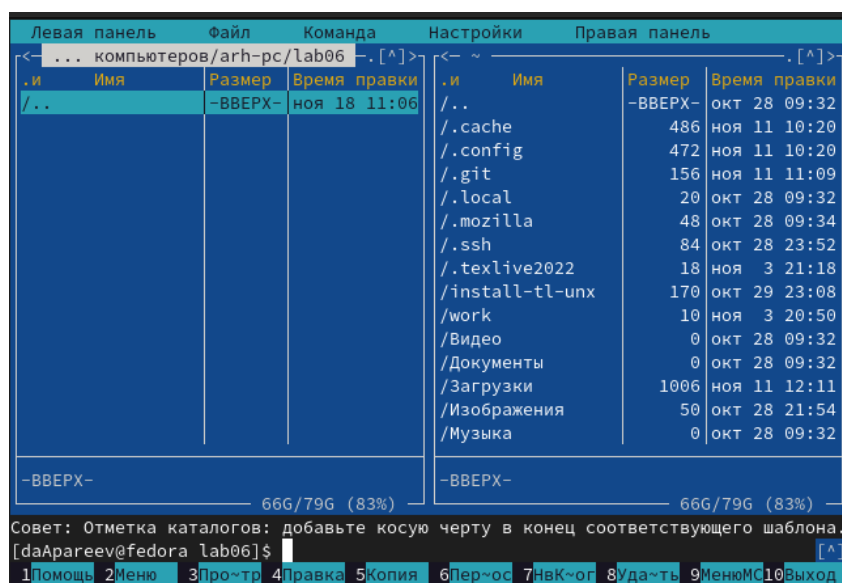


Рис. 3.2: 3

В строке ввода прописываю команду `touch lab6-1.asm`, чтобы создать файл, в котором буду работать (рис. 3.3).

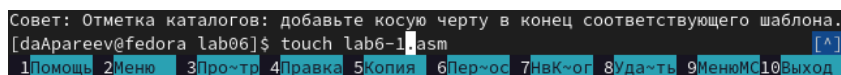


Рис. 3.3: 4

С помощью функциональной клавиши F4 открываю созданный файл для редактирования в редакторе. Ввожу в файл код программы для запроса строки у пользователя (рис. 3.4). Далее выхожу из файла, сохраняя изменения (рис. 3.4).

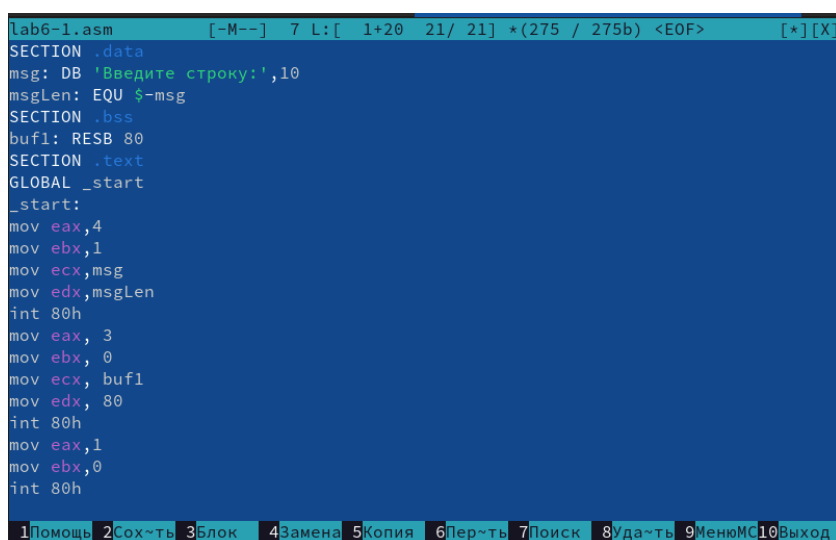



Рис. 3.4: 5

С помощью функциональной клавиши F3 открываю файл для просмотра, чтобы проверить, содержит ли файл текст программы (рис. 6 3.5).

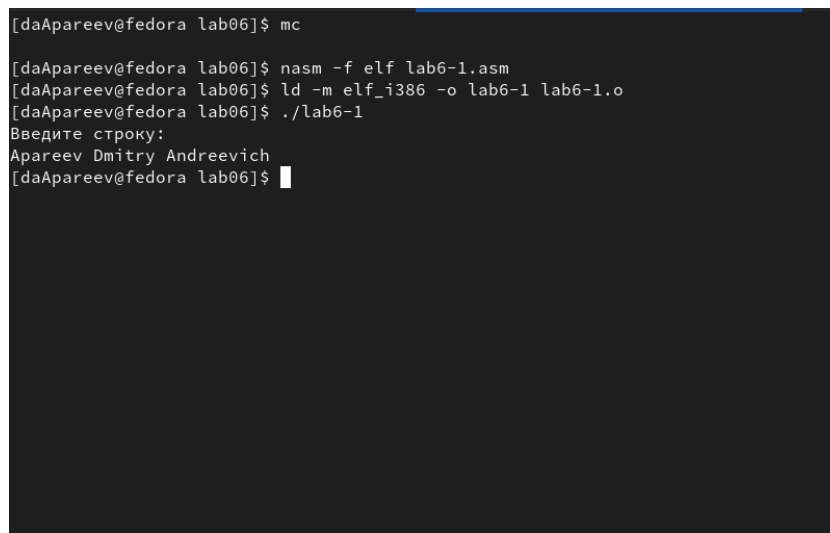


```
/home/daApareev/work/st~rh-pc/lab06/lab6-1.asm 275/275 100%
SECTION .data
msg: DB 'Введите строку:',10
msgLen: EQU $-msg
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax, 3
mov ebx, 0
mov ecx, buf1
mov edx, 80
int 80h
mov eax,1
mov ebx,0
int 80h

1Помощь 2Раз-рн 3Выход 4Hex 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Рис. 3.5: 6

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-1.asm`. Создался объектный файл `lab6-1.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-1 lab6-1.o`. Создался исполняемый файл `lab6-1`. Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу



```
[daApareev@fedora lab06]$ mc
[daApareev@fedora lab06]$ nasm -f elf lab6-1.asm
[daApareev@fedora lab06]$ ld -m elf_i386 -o lab6-1 lab6-1.o
[daApareev@fedora lab06]$ ./lab6-1
Введите строку:
Apareev Dmitry Andreevich
[daApareev@fedora lab06]$
```

Рис. 3.6: 7

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab06

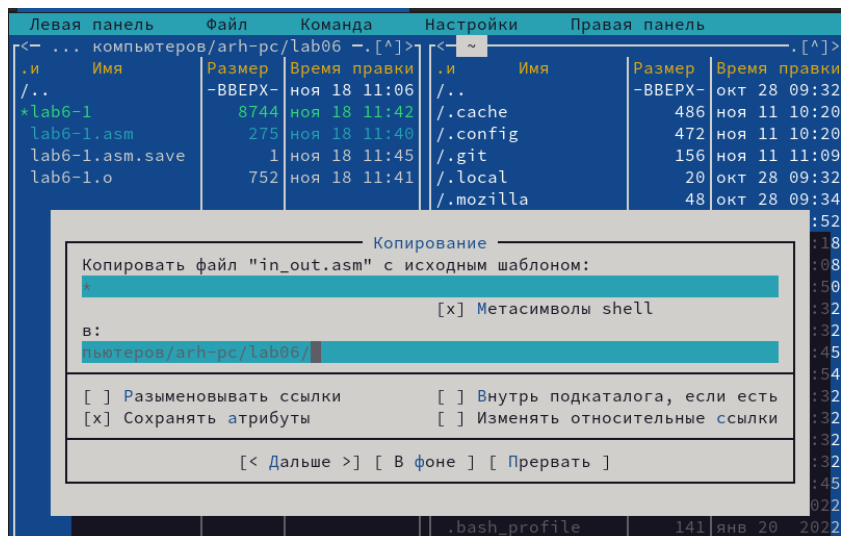


Рис. 3.7: 8

С помощью функциональной клавиши F5 копирую файл lab6-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис.9 ??).

!(image/9.png){ #fig:009 width=70% }

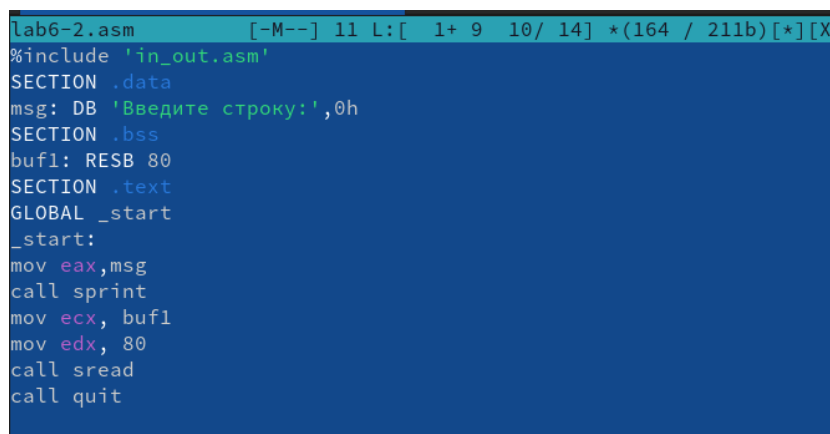
Изменяю содержимое файла lab6-2.asm во встроенном редакторе nano, чтобы в программе использовались подпрограммы из внешнего файла in_out.asm. Транслирую текст программы файла в объектный файл командой `nasm -f elf lab6-2.asm`. Создался объектный файл lab6-2.o. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab6-2 lab6-2.o` Создался исполняемый файл lab6-2. Запускаю исполняемый файл (рис. 3.8).

```
[daApareev@fedora lab06]$ nasm -f elf lab6-2.asm
[daApareev@fedora lab06]$ ld -m elf_i386 -o lab6-2 lab6-2.o
[daApareev@fedora lab06]$ ./lab6-2
Введите строку:
Apareev Dmitry Andreevich
```

Рис. 3.8: 10

Открываю файл lab6-2.asm для редактирования в nano функциональной клави-

шей F4. Изменяю в нем подпрограмму sprintLF на sprint. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. 3.9).



```
lab6-2.asm [-M--] 11 L: [ 1+ 9 10/ 14] *(164 / 211b)[*][X]
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
call quit
```

Рис. 3.9: 11

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл. Разница между первым исполняемым файлом lab6-2 и вторым lab6-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая выполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами sprintLF и sprint.

Открываю файл report.md с помощью любого текстового редактора gedit. Компилирую файл с отчетом. Загружаю отчет на GitHub.

4 Задания для самостоятельной работы

Создаю копию файла lab6-1.asm с именем lab6-1-1.asm с помощью функциональной клавиши F(рис.4.1)

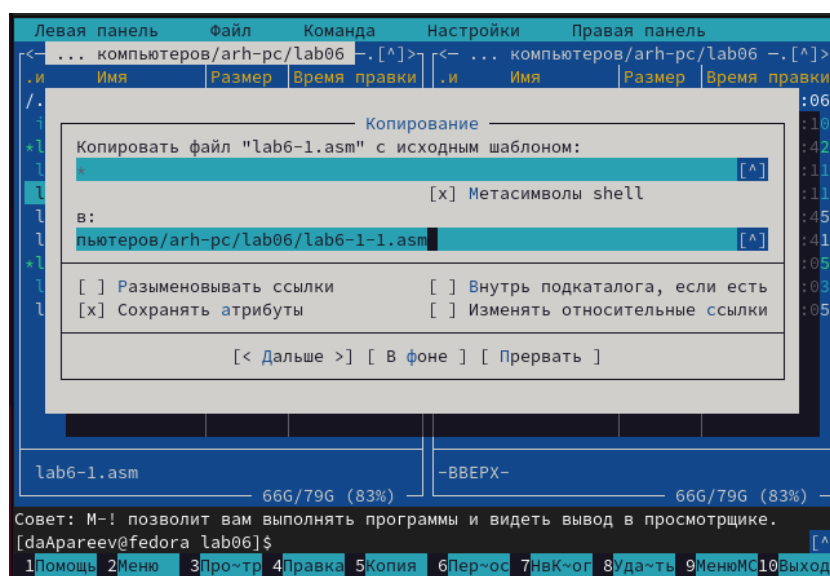
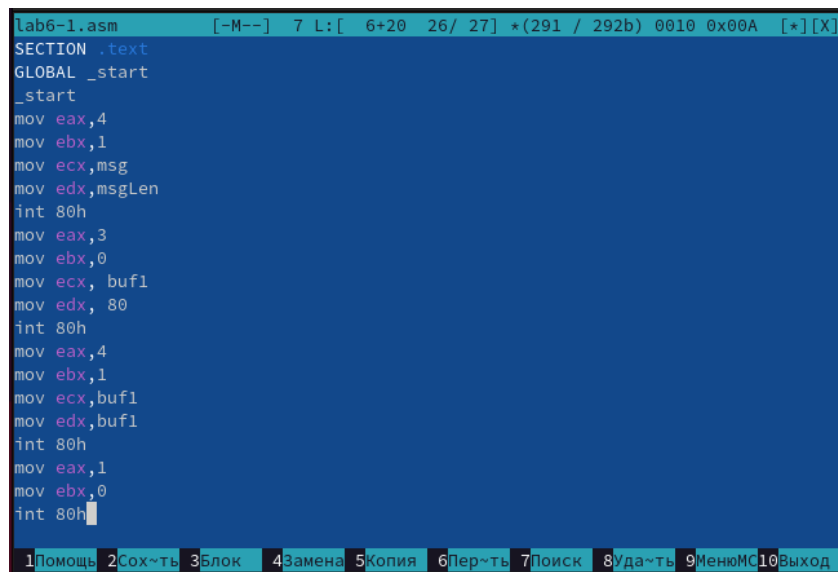


Рис. 4.1: 12

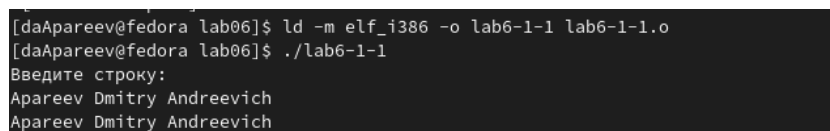
С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис.4.2).



```
lab6-1.asm [-M--] 7 L: [ 6+20 26/ 27] *(291 / 292b) 0010 0x00A [*][X]
SECTION .text
GLOBAL _start
_start
mov eax,4
mov ebx,1
mov ecx,msg
mov edx,msgLen
int 80h
mov eax,3
mov ebx,0
mov ecx,buf1
mov edx,80
int 80h
mov eax,4
mov ebx,1
mov ecx,buf1
mov edx,buf1
int 80h
mov eax,1
mov ebx,0
int 80h
1Помощь 2Сох-ть 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 4.2: 13

Создаю объектный файл lab6-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab6-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис.4.3).



```
[daApareev@fedora lab06]$ ld -m elf_i386 -o lab6-1-1 lab6-1-1.o
[daApareev@fedora lab06]$ ./lab6-1-1
Введите строку:
Apareev Dmitry Andreevich
Apareev Dmitry Andreevich
```

Рис. 4.3: 14

Создаю копию файла lab6-2.asm с именем lab6-2-1.asm с помощью функциональной клавиши F5 (рис.154.4).

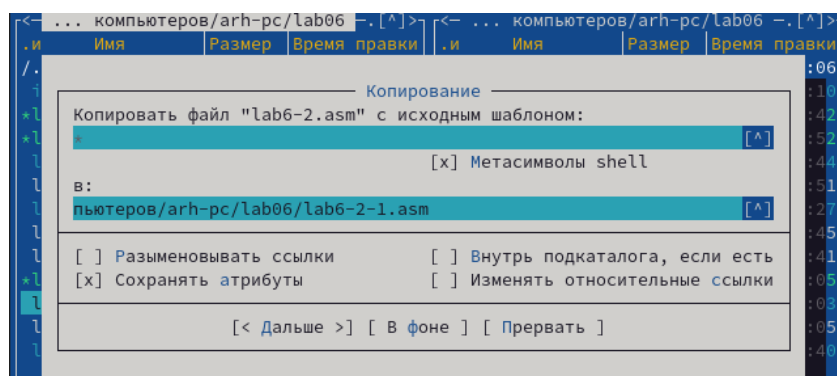


Рис. 4.4: 15

С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. 4.5).

```
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите строку:',0h
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax,msg
call sprint
mov ecx, buf1
mov edx, 80
call sread
mov eax,4
mov ebx,1
mov ecx,buf1
int 80h
call quit
```

Рис. 4.5: 16

Создаю объектный файл lab6-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab6-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные рис. 4.6).

```
[daApareev@fedora lab06]$ ld -m elf_i386 -o lab6-2-1 lab6-2-1.o
[daApareev@fedora lab06]$ ./lab6-2-1
Введите строку: Apareev Dmitry Andreevich
Apareev Dmitry Andreevich
```

Рис. 4.6: 17

Добавляю файлы в git , Сохраняю файлы в git , Отправляю файлы на сервер

5 Вывод

При выполнении данной лабораторной работы я приобрел практические навыки работы в Midnight Commander, а также освоил инструкции языка ассемблера `mov` и `int`.