

Отчёт по лабораторной работе №5

дисциплина: Архитектура компьютера

Бражко Александра Александровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выполнение заданий для самостоятельной работы	14
6	Выводы	18
	Список литературы	19

Список иллюстраций

4.1	Использование команды <code>mc</code>	8
4.2	Переход в каталог и создание нового	9
4.3	Создание файла	9
4.4	Использование редактора <code>mcedit</code>	10
4.5	Проверка текста в файле	10
4.6	Оттранслирование текста	11
4.7	Компоновка и ввод ФИО	11
4.8	Скопированный файл	12
4.9	Использование команды <code>mc</code>	12
4.10	Изменяем текст программы	13
4.11	Создание и проверка файла	13
5.1	Создание копии	14
5.2	Редактирование текста программы в файле	15
5.3	Выполнение ряда программ	15
5.4	Создание копии	16
5.5	Редактирование текста программы в файле	16
5.6	Выполнение ряда программ	17

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . . .	7
-----	---	---

1 Цель работы

Целью данной лабораторной работы - приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Задание

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

3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 3.1 приведено краткое описание стандартных каталогов Unix.

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
/	Корневая директория, содержащая всю файловую систему
/bin	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
/etc	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
/home	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
/media	Точки монтирования для сменных носителей
/root	Домашняя директория пользователя root
/tmp	Временные файлы
/usr	Вторичная иерархия для данных пользователя

Более подробно про Unix см. в [1–4].

4 Выполнение лабораторной работы

Открываем Midnight Commander, используя команду `mc`. (рис. 4.1).

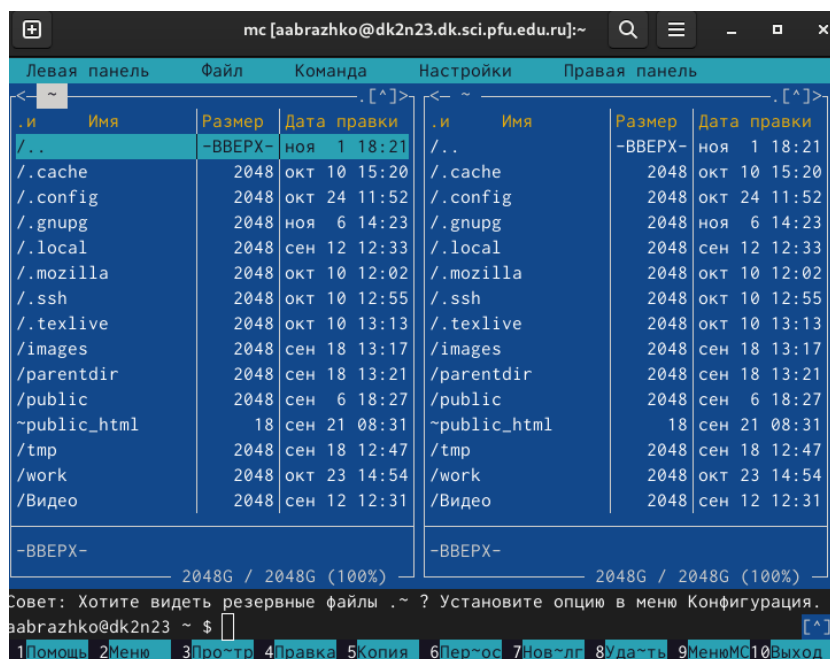


Рис. 4.1: Использование команды `mc`

Переходим в каталог `~/work/arch-рс`, созданный при выполнении лабораторной работы №4, и создаём новый каталог с именем `lab05` (рис. 4.2).

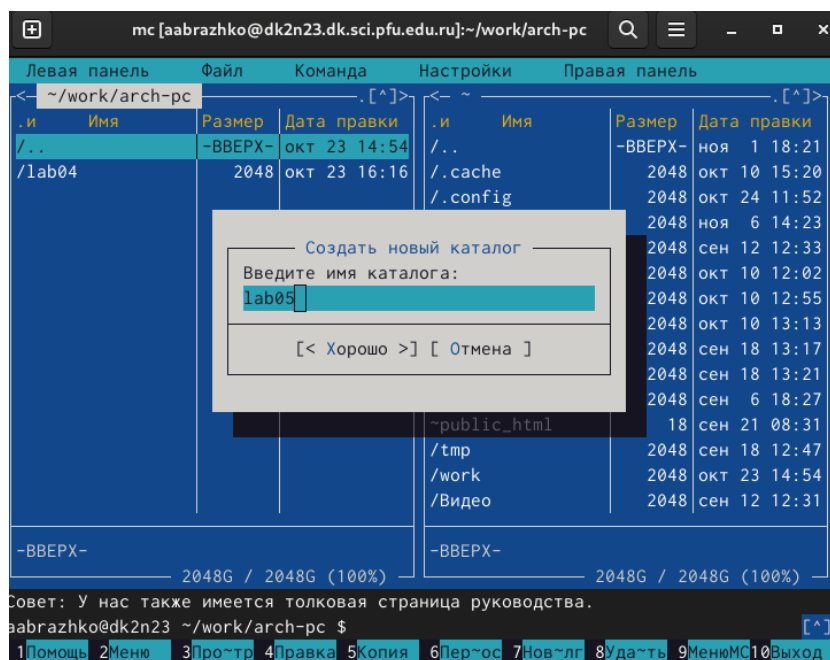


Рис. 4.2: Переход в каталог и создание нового

Используя команду touch, создаём файл lab5-1.asm (рис. 4.3).

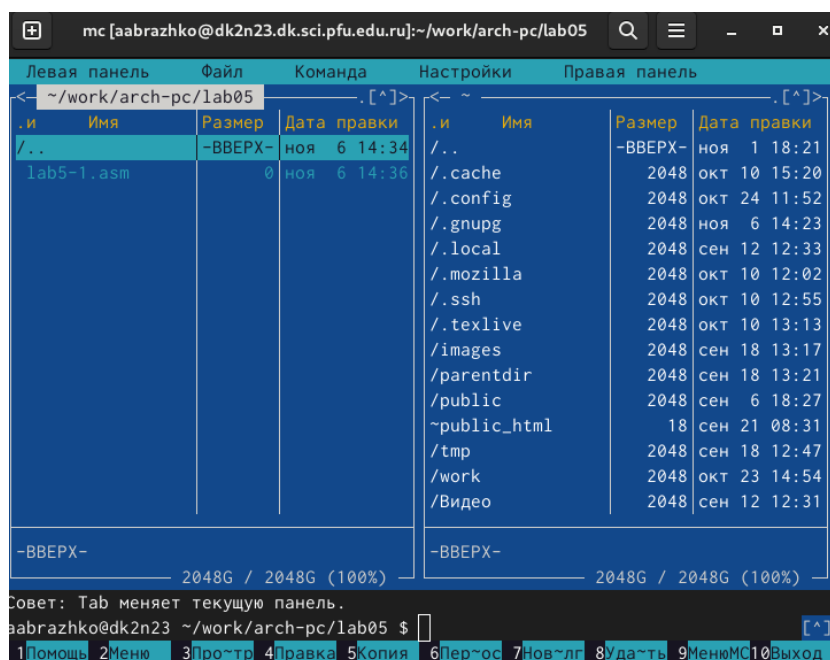
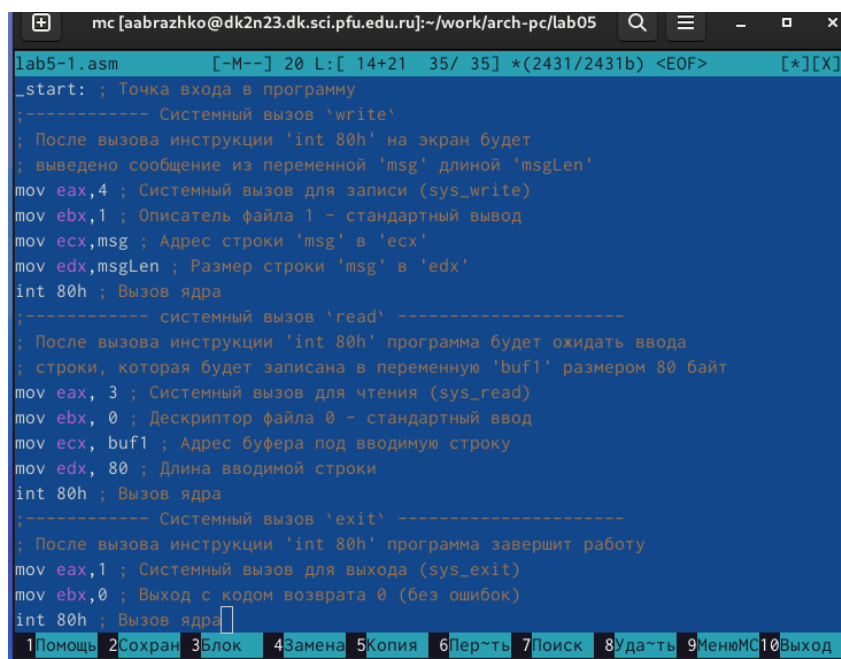


Рис. 4.3: Создание файла

С помощью клавиши F4 открываем файл lab5-1.asm для редактирования во

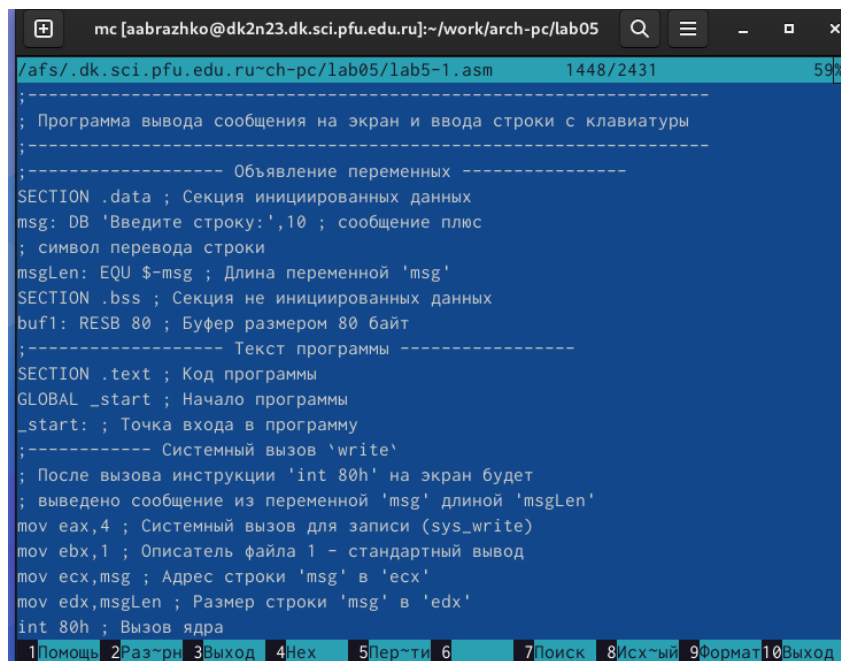
встроенном редакторе. Используем редактор mcedit (рис. 4.4).



```
mc [aabrazhko@dk2n23.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab05
lab5-1.asm [-M--] 20 L: 14+21 35/ 35] *(2431/2431b) <E0F> [*][X]
_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
;----- 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
1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюMC10Выход
```

Рис. 4.4: Использование редактора mcedit

Вводим текст программы из листинга 5.1, сохраняем и закрываем. С помощью клавиши F3 открываем файл для просмотра (рис. 4.5).



```
mc [aabrazhko@dk2n23.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab05
/afs/.dk.sci.pfu.edu.ru~ch-pc/lab05/lab5-1.asm 1448/2431 59%
;-----
; 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
1Помощь 2Разрешить 3Выход 4Нех 5Перейти 6 7Поиск 8Исх-ый 9Формат10Выход
```

Рис. 4.5: Проверка текста в файле

Оттранслируем текст программы в объектный файл. (рис. 4.6).

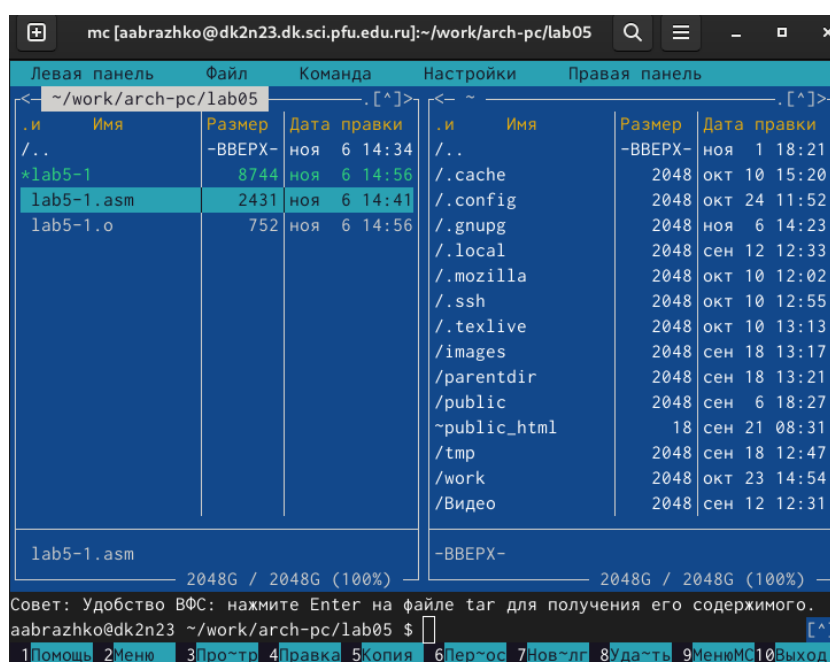


Рис. 4.6: Оттранслирование текста

Выполняем компоновку объектного файла и запускаем получившийся исполняемый файл. После вывода строки 'Введите строку:' вводим своё ФИО (рис. 4.7).

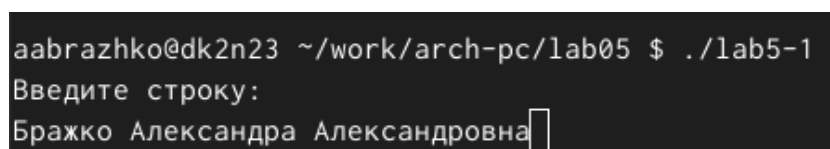
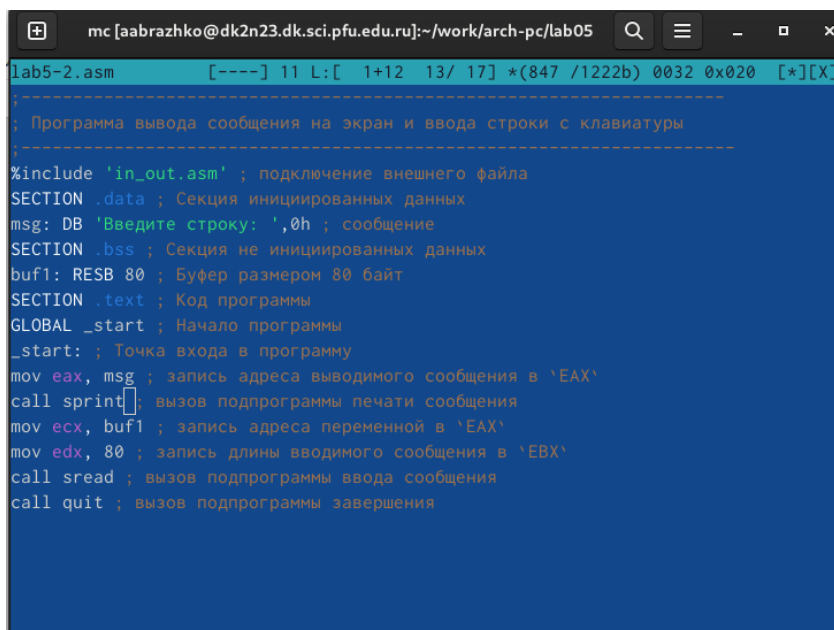


Рис. 4.7: Компоновка и ввод ФИО

Скачиваем файл in_out.asm со страницы курса ТУИС и копируем в созданный каталог lab05 (рис. 4.8).

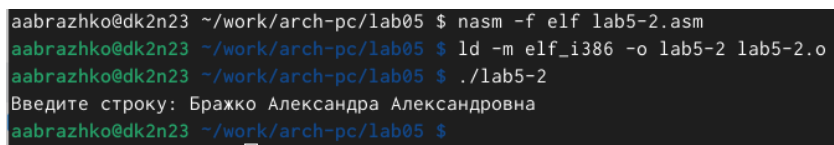
грамм из внешнего файла `in_out.asm` в соответствии с листингом 5.2. Заменяем подпрограмму `sprintLF` на `sprint` (рис. 4.10).



```
mc [aabrazhko@dk2n23.dk.sci.pfu.edu.ru]:~/work/arch-pc/lab05
lab5-2.asm  [----] 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 иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных 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 sprint ; 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. 4.10: Изменяем текст программы

Создаём исполняемый файл и проверяем его работу (рис. 4.11).



```
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Бражко Александра Александровна
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $
```

Рис. 4.11: Создание и проверка файла

5 Выполнение заданий для самостоятельной работы

Создём копию файла lab5-1.asm с именем lab5-1-1.asm с помощью клавиши F5 (в приведённом ниже рисунке показан файл в конечном виде, то есть копия+объектный файл+скомпонованный) (рис. 5.1).

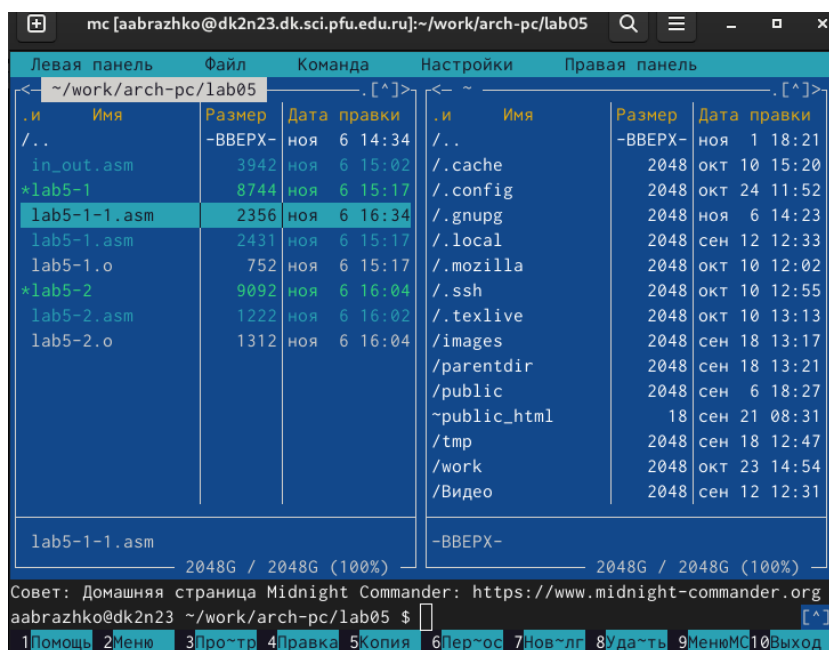
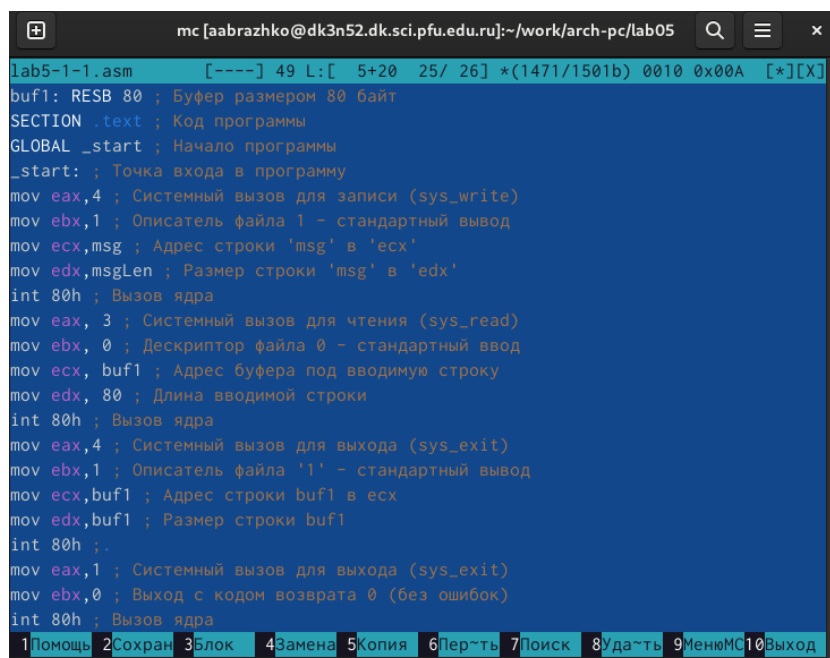


Рис. 5.1: Создание копии

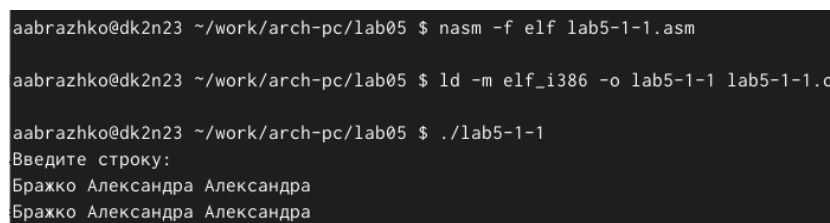
С помощью клавиши F4 открываю созданный файл для редактирования. Меняю программу, так чтобы она выводила приглашение типа 'Введите строку', вводим своё ФИО, выводила введенную строку на экран (рис. 5.2).



```
lab5-1-1.asm  [----] 49 L: [ 5+20 25/ 26] *(1471/1501b) 0010 0x00A [*][X]
buf1: RESB 80 ; Буфер размером 80 байт
SECTION .text ; Код программы
GLOBAL _start ; Начало программы
_start: ; Точка входа в программу
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
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 для выхода (sys_exit)
mov ebx,1 ; Описатель файла '1' - стандартный вывод
mov ecx,buf1 ; Адрес строки buf1 в ecx
mov edx,buf1 ; Размер строки buf1
int 80h ;
mov eax,1 ; Системный вызов для выхода (sys_exit)
mov ebx,0 ; Выход с кодом возврата 0 (без ошибок)
int 80h ; Вызов ядра
1Помощь 2Сохран 3Блок 4Замена 5Копия 6Пер-ть 7Поиск 8Уда-ть 9МенюМС10Выход
```

Рис. 5.2: Редактирование текста программы в файле

Создание объектного файла, компоновка, запуск полученного исполняемого файла (рис. 5.3).



```
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку:
Бражко Александра Александра
Бражко Александра Александра
```

Рис. 5.3: Выполнение ряда программ

Создание копии файла lab5-2.asm, называем lab5-2-1.asm (в приведённом ниже рисунке показан файл в конечном виде, то есть копия+объектный файл+скомпонованный) (рис. 5.4).

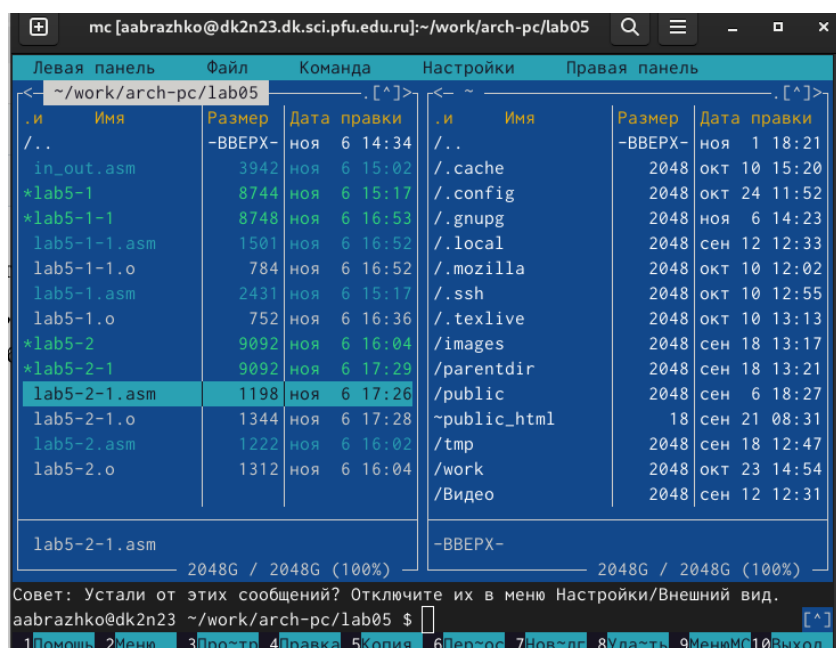


Рис. 5.4: Создание копии

С помощью клавиши F4 открываю созданный файл для редактирования. Меняю программу, так чтобы она выводила приглашение типа 'Введите строку', вводим своё ФИО, выводила введённую строку на экран (рис. 5.5).

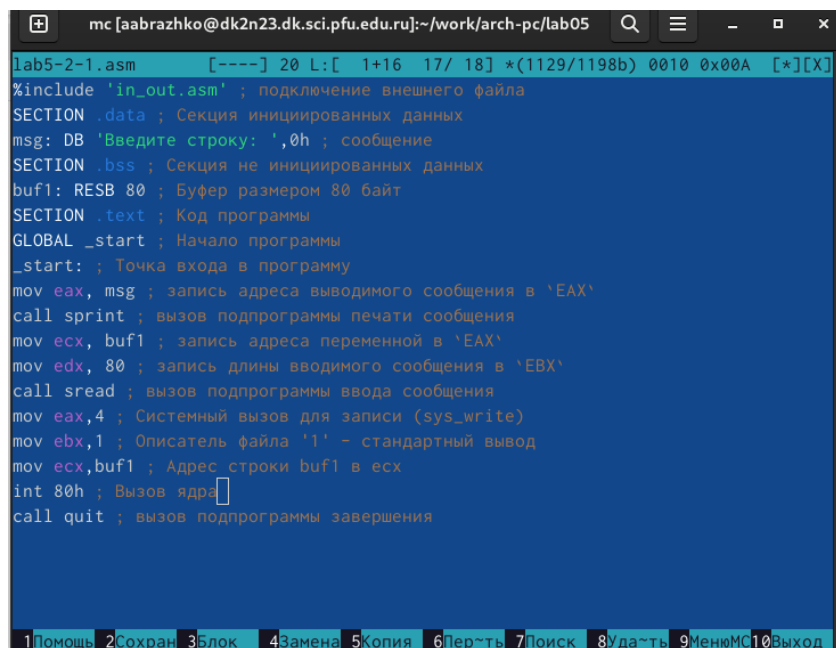


Рис. 5.5: Редактирование текста программы в файле

Создание объектного файла, компоновка, запуск полученного исполняемого файла (рис. 5.6).

```
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-1.asm
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
aabrazhko@dk2n23 ~/work/arch-pc/lab05 $ ./lab5-2-1
Введите строку: Бражко Александра Александра
Бражко Александра Александра
```

Рис. 5.6: Выполнение ряда программ

6 Выводы

Здесь кратко описываются итоги проделанной работы.

Список литературы

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.
2. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.