Лабораторная работа №5

Архитектура компьютера

Казначеева Кристина Никитична

Содержание

1 Цель работы

Целью лабораторной работы является приобретение практических навыков использования файлового менеджера Midnight Commander и освоение основных инструкций языка ассемблера “mov” и “int”.

2 Задание

Эта лабораторная работа направлена на освоение программ, выводящих сообщение на экран и считывающих ввод с клавиатуры. Мы познакомимся с двумя вариантами: • Программирование с использованием файла in_out.asm: изучение принципов подключения внешнего файла с готовыми функциями ввода/вывода. • Программирование без внешнего файла: реализация функции вывода и ввода, используя системные вызовы.

3 Выполнение лабораторной работы

Открываем Midnight Commander (рис. 3.1).

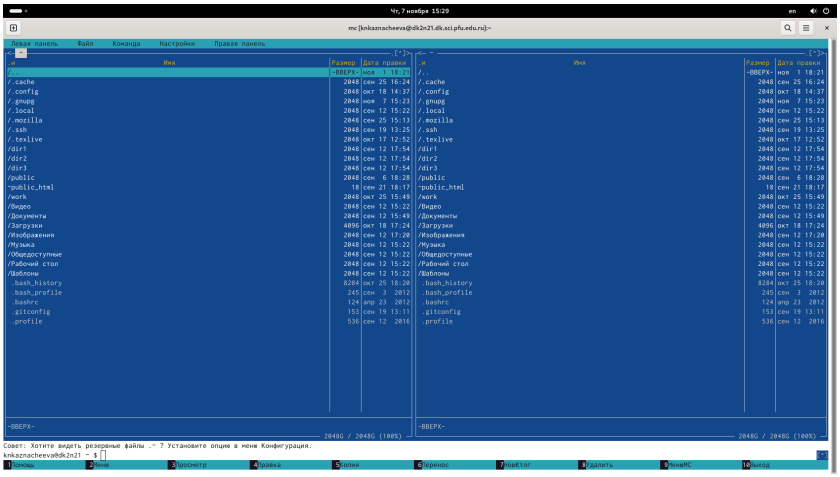

Рис. 3.1: Открытие Midnight Commander

Перейдём в каталог ~/work/arch-pc, созданный при выполнении лабораторной работы №4 (рис. 3.2).

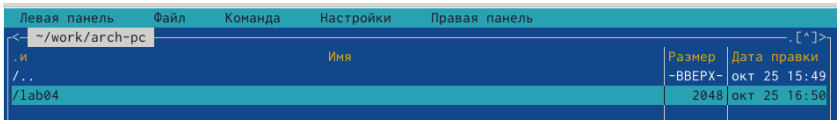

Рис. 3.2: Переход в каталог

Создаём папку lab05 и переходим в созданный каталог,затем. пользуясь строкой ввода и командой touch, создаём файл lab5-1.asm (рис. 3.3).

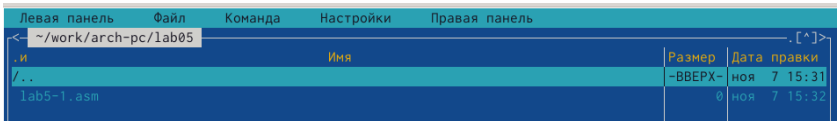

Рис. 3.3: Создание папки и файла

Открываем файл lab5-1.asm для редактирования во встроенном редакторе и вводим текст программы (рис. 3.4).

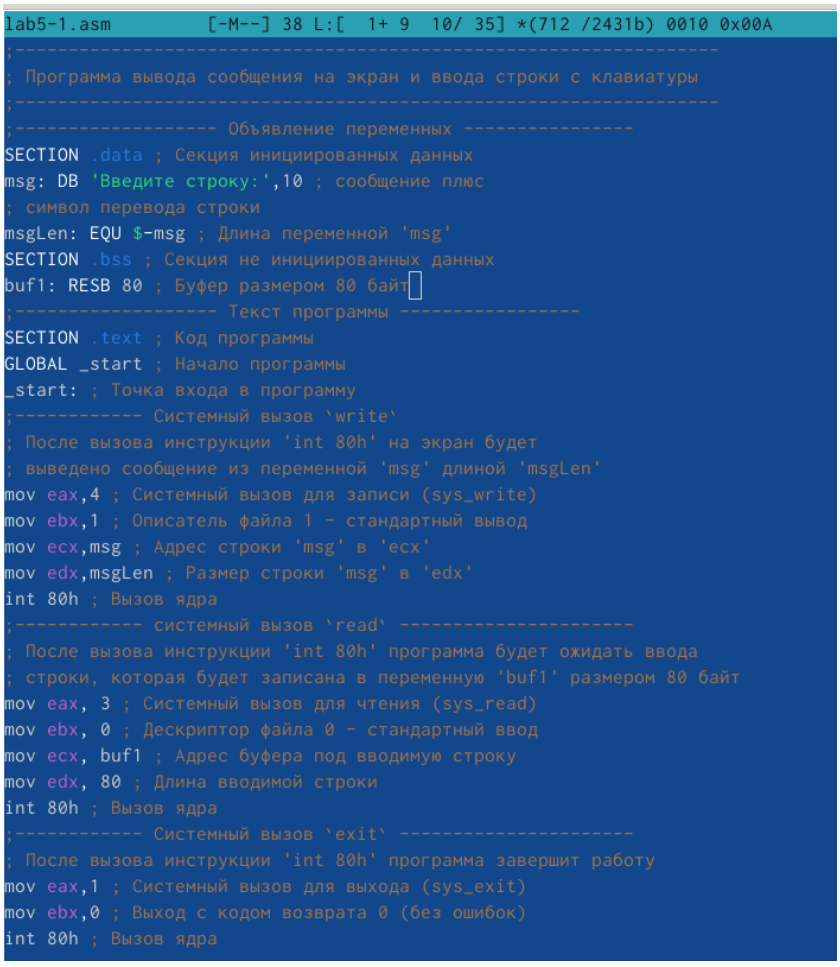

Рис. 3.4: Ввод текста программы

Открываем файл lab5-1.asm для просмотра, чтобы убедиться, что файл содержит текст программы(рис. 3.5).

```
/afs/.dk.sci.pfu.edu.ru/home/k/n/knkaznacheeva/work/arch-pc/lab05/lab5-1.asm
;-----------------------------------------------------------------
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----------------------------------------------------------------
;------------------- Объявление переменных ----------------
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss ; Секция не инициированных 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
;------------ 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. 3.5: Открытие файла

Оттранслируем текст программы lab5-1.asm в объектный файл и выполним компоновку объектного файла, затем запустим получившийся исполняемый файл. Программа выводит строку ‘Введите строку:’ и ожидает ввода с клавиатуры (рис. 3.6).

```
knkaznacheeva@dk2n21 ~ $ mc

knkaznacheeva@dk2n21 ~ $ mc

knkaznacheeva@dk2n21 ~ $ nasm -f elf lab5-1.asm
nasm: fatal: unable to open input file `lab5-1.asm' No such file or director
knkaznacheeva@dk2n21 ~ $ ld -m elf_i386 -o lab5-1 lab5-1.o
ld: невозможно найти lab5-1.o: Нет такого файла или каталога
knkaznacheeva@dk2n21 ~ $ cd work/arch-pc/lab05/
knkaznacheeva@dk2n21 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1.asm
knkaznacheeva@dk2n21 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1 lab5-1.
knkaznacheeva@dk2n21 ~/work/arch-pc/lab05 $ ./lab5-1
Введите строку:
Казначеева Кристина Никитична
knkaznacheeva@dk2n21 ~/work/arch-pc/lab05 $
```

Рис. 3.6: Запуск файла

В одной из панелей mc открываем каталог с файлом lab5-1.asm. В другой панели каталог со скаченным файлом in_out.asm. Скомпилируем файл in_out.asm в каталог с файлом lab5-1.asm (рис. 3.7).

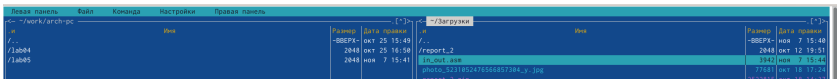

Рис. 3.7: Компиляция файла

Создаём копию файла lab5-1.asm с именем lab5-2.asm (рис. 3.8):

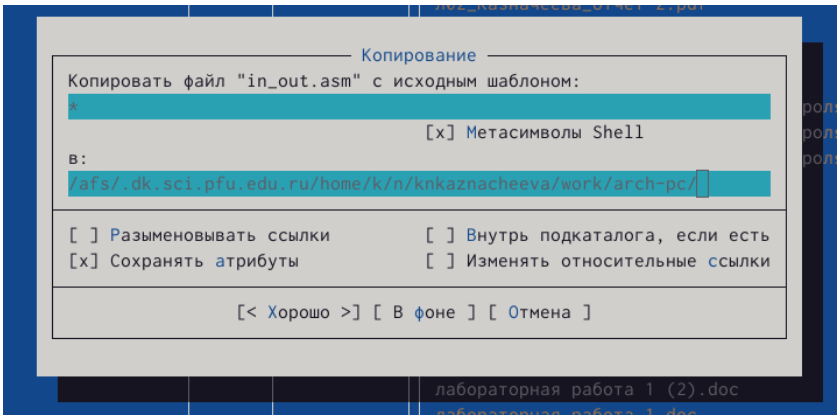

Рис. 3.8: Создание копии файла

Исправляем текст программы в файле lab5-2.asm с использованием подпрограмм из внешнего файла in_out.asm и создаём исполняемый файл (рис. 3.9).

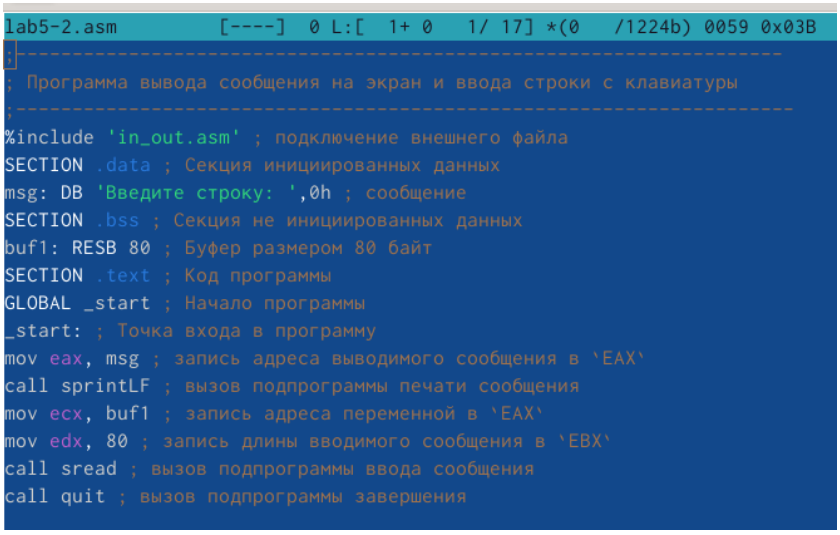

Рис. 3.9: Создание файла

Проверяем работу файла (рис. 3.10).

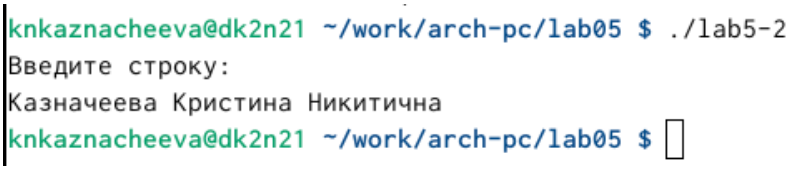

Рис. 3.10: Проверка работы файла

В файле lab5-2.asm заменяем подпрограмму sprintLF на sprint и создаём исполняемый файл (рис. 3.11).

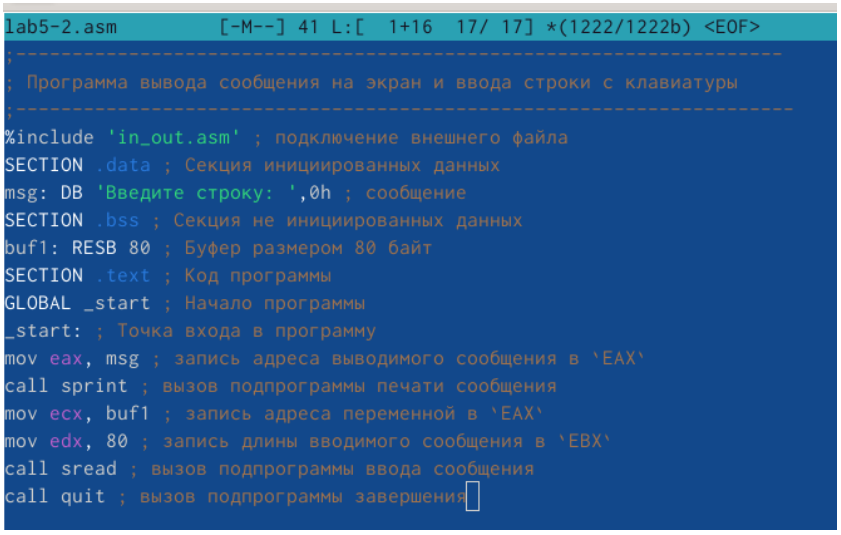

Рис. 3.11: Замена подпрограммы sprintLF на sprint

Проверяем работу файла (рис. 3.12).

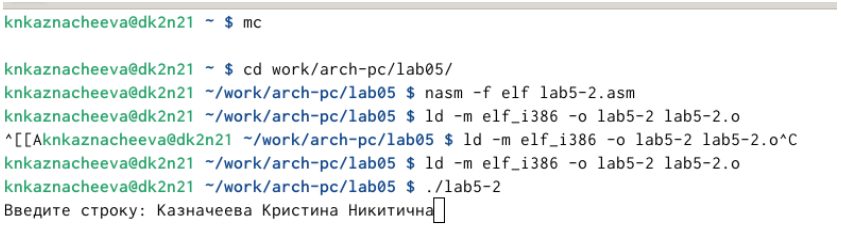

Рис. 3.12: Проверка работы файла

Создаём копию файла lab5-1.asm и вносим 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 (рис. 3.13).

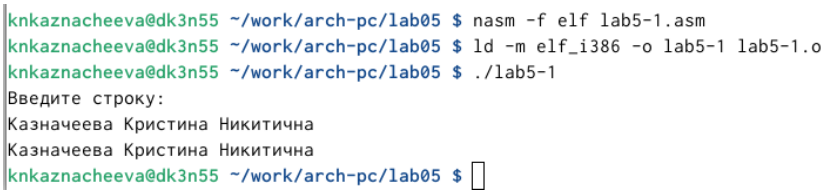

Рис. 3.13: Запуск получившегося файла

Создаём копию файла lab5-2.asm и исправляем текст программы с использо-

вание 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 (рис. 3.14).

```
knkaznacheeva@dk3n55 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
knkaznacheeva@dk3n55 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2 lab5-2.o
knkaznacheeva@dk3n55 ~/work/arch-pc/lab05 $ ./lab5-2
Введите строку: Казначеева Кристина Никитична
Казначеева Кристина Никитична
knkaznacheeva@dk3n55 ~/work/arch-pc/lab05 $
```

Рис. 3.14: Запуск получившегося файла

4 Выводы

В ходе работы мы успешно освоили несколько ключевых навыков:

1. Работа с файловым менеджером: Мы получили практический опыт работы с Midnight Commander, научились эффективно управлять файлами и директориями.

2. Основы ассемблера: Мы изучили основные инструкции языка ассемблера “mov” и “int”, что позволяет нам перемещать данные в регистры и вызывать системные функции.

3. Программирование ввода/вывода: Мы освоили создание программ, которые выводят сообщения на экран и считывают данные с клавиатуры.

4. Подключение внешних файлов: Мы познакомились с двумя подходами к реализации функций ввода/вывода: • Использование готового кода: Мы изучили принципы подключения внешнего файла in_out.asm, который предоставляет готовые функции ввода/вывода. • Самостоятельная реализация: Мы освоили необходимые системные вызовы, чтобы самостоятельно реализовать функции ввода/вывода.

В целом, мы получили ценный практический опыт в работе с ассемблером и файловым менеджером Midnight Commander, что позволит нам уверенно решать разнообразные задачи в будущем.