

Отчёт по лабораторной работе №6

дисциплина: Архитектура компьютера

Студент: Попов Н.Ю. Группа: НММбд-03-22

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Выводы	10

1 Цель работы

Целью моей работы является приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера `mov` и `int`.

2 Выполнение лабораторной работы

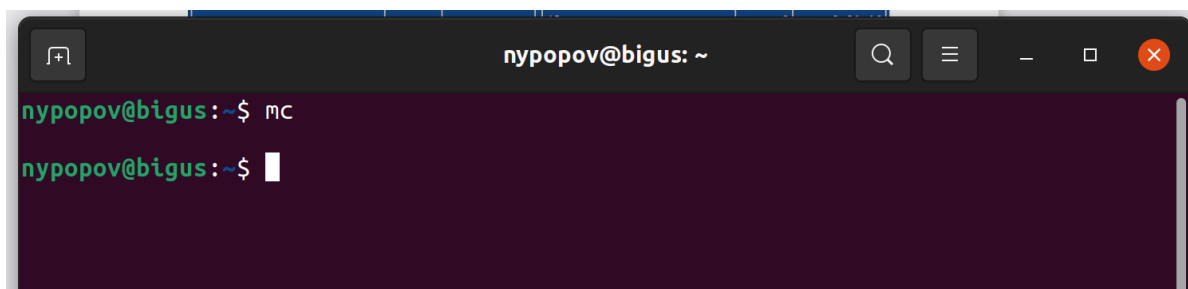


Рис. 2.1: Запуск Midnight Commander

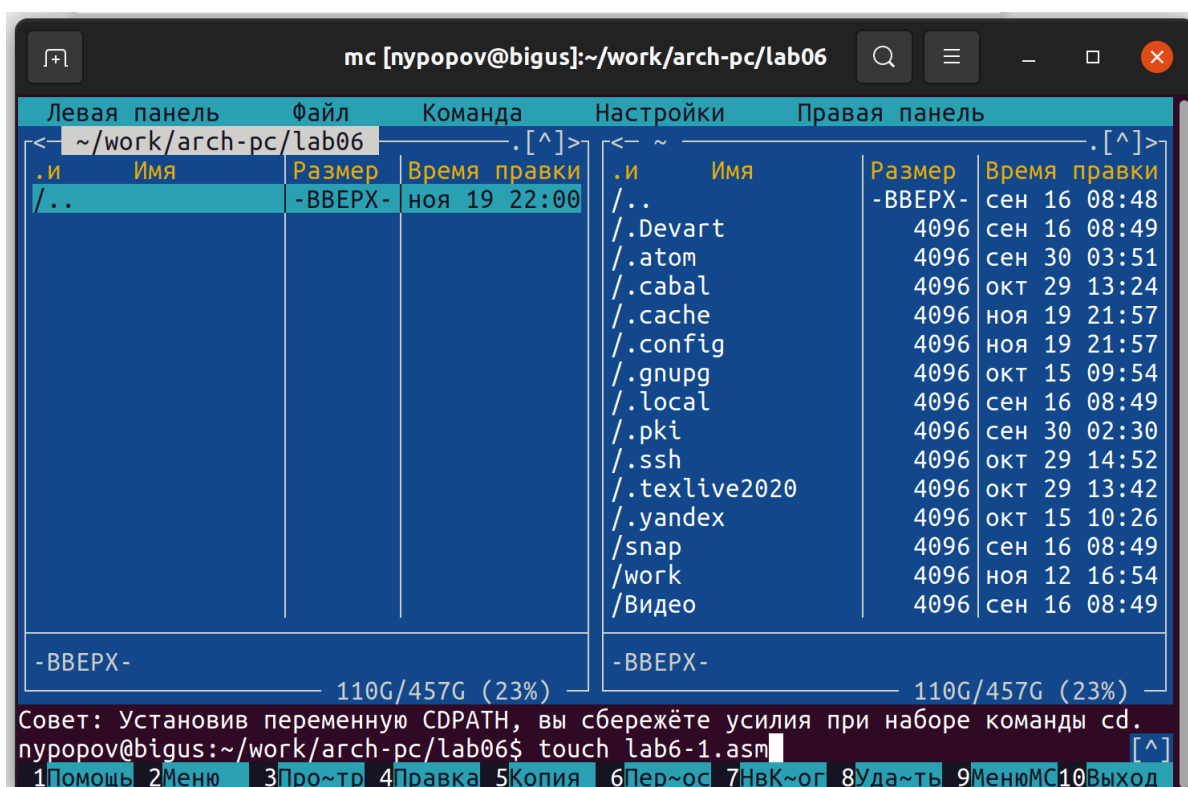


Рис. 2.2: Создание каталога

```

nypopov@bigus:~/work/arch-pc/lab06$ touch lab6-1.asm

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/mcedit
 4. /usr/bin/vim.tiny
 5. /usr/bin/code
 6. /bin/ed

Choose 1-6 [1]: █

```

Рис. 2.3: Создание файла

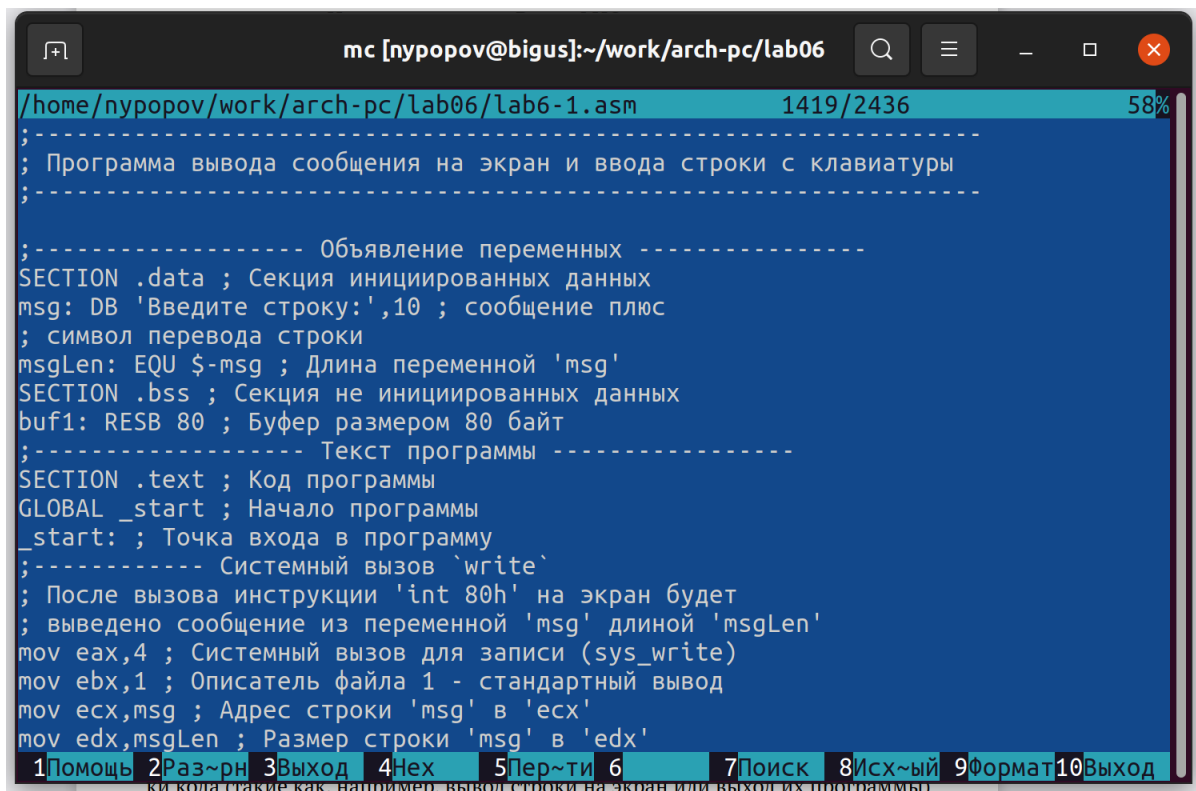
```

GNU nano 5.6.1 /home/nypopov/work/arch-pc/lab06/lab6-1.asm *
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 ; Файловый дескриптор 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
█

```

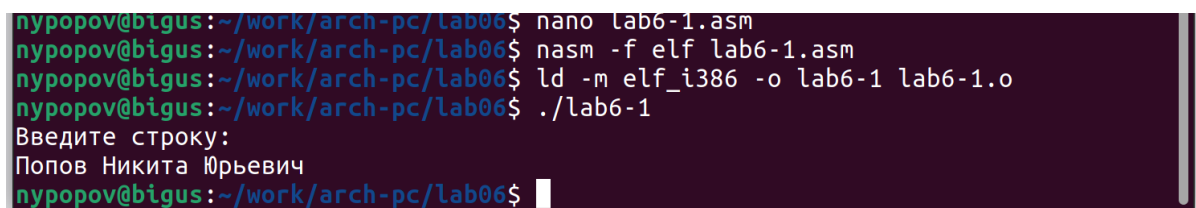
[^]G Справка [^]O Записать [^]W Поиск [^]K Вырезать [^]T Выполнить [^]C Позиция
[^]X Выход [^]R ЧитФайл [^]\ Замена [^]U Вставить [^]J Выводить [^]↵ К строке

Рис. 2.4: Ввод текста программы



```
mc [nyporov@bigus]:~/work/arch-pc/lab06
/home/nyporov/work/arch-pc/lab06/lab6-1.asm 1419/2436 58%
;
; 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'
1Помощь 2Раз~рн 3Выход 4Нех 5Пер~ти 6 7Поиск 8Исх~ый 9Формат10Выход
```

Рис. 2.5: Проверка содержимого файла



```
nyporov@bigus:~/work/arch-pc/lab06$ nano lab6-1.asm
nyporov@bigus:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
nyporov@bigus:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
nyporov@bigus:~/work/arch-pc/lab06$ ./lab6-1
Введите строку:
Попов Никита Юрьевич
nyporov@bigus:~/work/arch-pc/lab06$
```

Рис. 2.6: Трансляция, компоновка и запуск

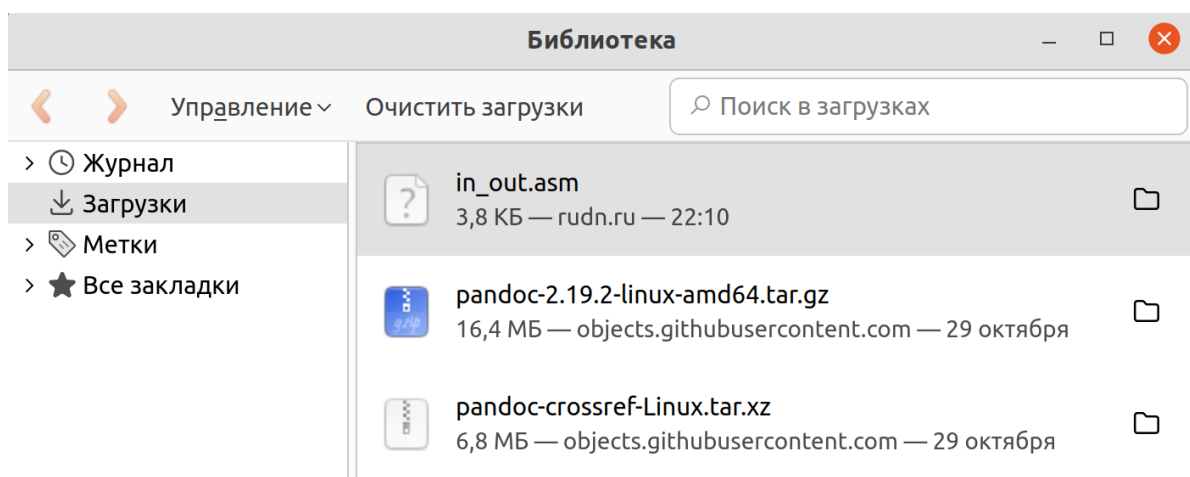


Рис. 2.7: Загрузка файла

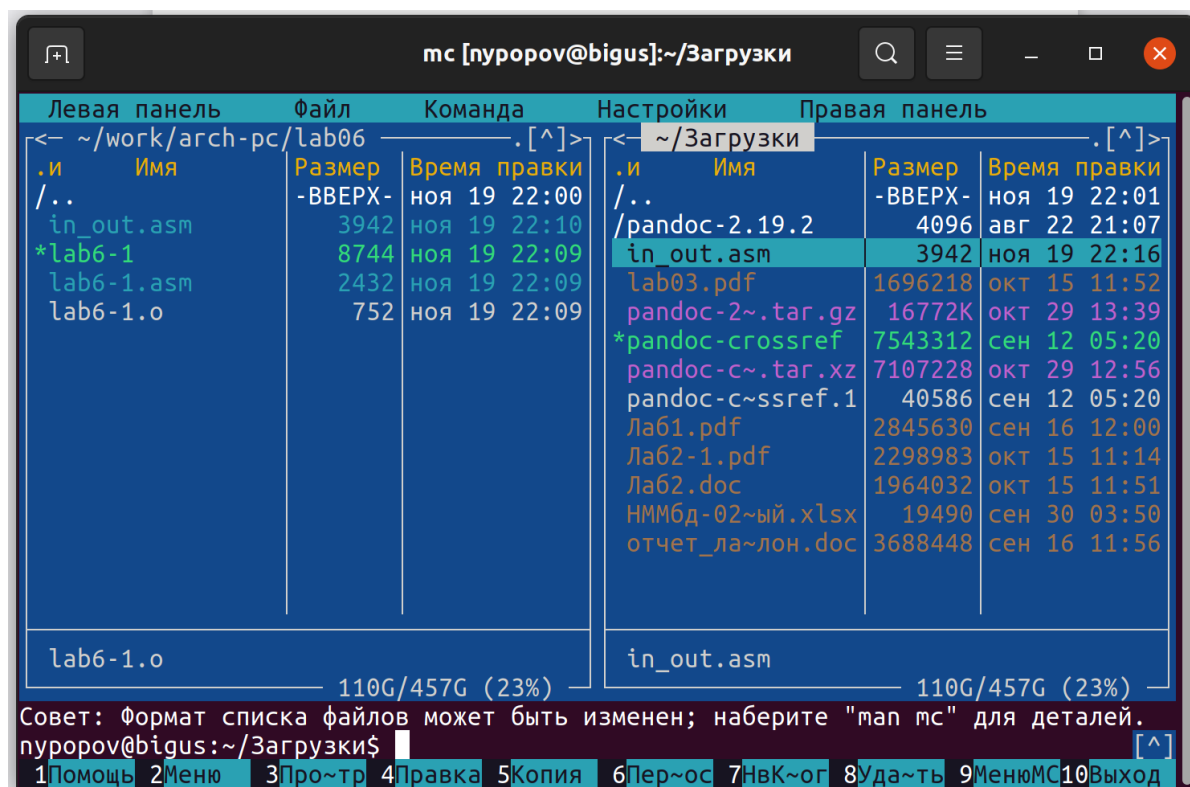


Рис. 2.8: Копирование файла

```

нуропов@bigus:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
нуропов@bigus:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
нуропов@bigus:~/work/arch-pc/lab06$ ./lab6-2
Введите строку:
Моя строка
нуропов@bigus:~/work/arch-pc/lab06$

```

Рис. 2.9: Проверка работы программы с использованием файла in_out.asm

```

GNU nano 5.6.1 lab6-2.asm *
;-----
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;-----
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не инициированных 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

^G Справка      ^O Записать     ^W Поиск       ^K Вырезать    ^T Выполнить   ^C Позиция
^X Выход        ^R ЧитФайл     ^\ Замена     ^U Вставить    ^J Выровнять   ^_ К строке

```

Рис. 2.10: Замена подпрограммы sprintf на sprint

```

нуропов@bigus:~/work/arch-pc/lab06$ nano lab6-2.asm
нуропов@bigus:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
нуропов@bigus:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
нуропов@bigus:~/work/arch-pc/lab06$ ./lab6-2
Введите строку: можно вводить в той же самой строке
нуропов@bigus:~/work/arch-pc/lab06$ строке
строке: команда не найдена
нуропов@bigus:~/work/arch-pc/lab06$

```

Рис. 2.11: Проверка работы программы с заменой


```
нуроров@bigus:~/work/arch-pc/lab06$ nano lab6-1-copy.asm
нуроров@bigus:~/work/arch-pc/lab06$ nasm -f elf lab6-1-copy.asm
нуроров@bigus:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1-copy lab6-1-copy.o
нуроров@bigus:~/work/arch-pc/lab06$ ./lab6-1-copy
Введите строку:
Ввожу строку и ожидаю её повторения
Ввожу строку и ожидаю её повторения
нуроров@bigus:~/work/arch-pc/lab06$
```

Рис. 2.12: Результат изменения первой программы

```
нуроров@bigus:~/work/arch-pc/lab06$ nano lab6-2-copy.asm
нуроров@bigus:~/work/arch-pc/lab06$ nasm -f elf lab6-2-copy.asm
нуроров@bigus:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2-copy lab6-2-copy.o
нуроров@bigus:~/work/arch-pc/lab06$ ./lab6-2-copy
Введите строку: ввожу строку и ожидаю её вывод
Ввожу строку и ожидаю её вывод
нуроров@bigus:~/work/arch-pc/lab06$
```

Рис. 2.13: Результат изменения второй программы

3 Выводы

Мною была приобретены практические навыки работы в Midnight Commander и освоены инструкции языка ассемблера mov и int.