

Отчет по лабораторной работе №5

Дисциплина: архитектура компьютера

Хрисанова Ксения Олеговна

Содержание

1	Цель работы	1
2	Задание.....	1
3	Теоретическое введение	1
4	Выполнение лабораторной работы.....	2
4.1	Основы работы с Midnight Commander.....	2
4.2	Работа в NASM	4
4.3	Подключение внешнего файла.....	5
4.4	Задание для самостоятельной работы	7
5	Выводы	7
	Список литературы	7

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 mc
2. Структура программы на языке ассемблера NASM
3. Подключение внешнего файла
4. Выполнение заданий для самостоятельной работы

3 Теоретическое введение

Midnight Commander (или просто mc) — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: секция кода программы (SECTION .text), секция инициализированных (известных во время компиляции) данных (SECTION .data) и секция неинициализированных данных (тех, под которые во время компиляции только отводится память, а значение

присваивается в ходе выполнения программы) (SECTION .bss). Для объявления инициированных данных в секции .data используются директивы DB, DW, DD, DQ и DT, которые резервируют память и указывают, какие значения должны храниться в этой памяти:

- DB (define byte) — определяет переменную размером в 1 байт; - DW (define word) — определяет переменную размером в 2 байта (слово); - DD (define double word) — определяет переменную размером в 4 байта (двойное слово); - DQ (define quad word) — определяет переменную размером в 8 байт (четвертое слово); - DT (define ten bytes) — определяет переменную размером в 10 байт. Директивы используются для объявления простых переменных и для объявления массивов. Для определения строк принято использовать директиву DB в связи с особенностями хранения данных в оперативной памяти. Инструкция языка ассемблера mov предназначена для дублирования данных источника в приёмнике.

```
mov dst,src
```

Здесь operand dst — приёмник, а src — источник. В качестве операнда могут выступать регистры (register), ячейки памяти (memory) и непосредственные значения (const). Инструкция языка ассемблера int предназначена для вызова прерывания с указанным номером.

```
int n
```

Здесь n — номер прерывания, принадлежащий диапазону 0–255. При программировании в Linux с использованием вызовов ядра sys_calls n=80h (принято задавать в шестнадцатеричной системе счисления).

4 Выполнение лабораторной работы

4.1 Основы работы с Midnight Commander

Введя соответствующую команду в терминале (рис. 1), я открываю Midnight Commander (рис. 2).



Рис. 1: Открытие Midnight Commander

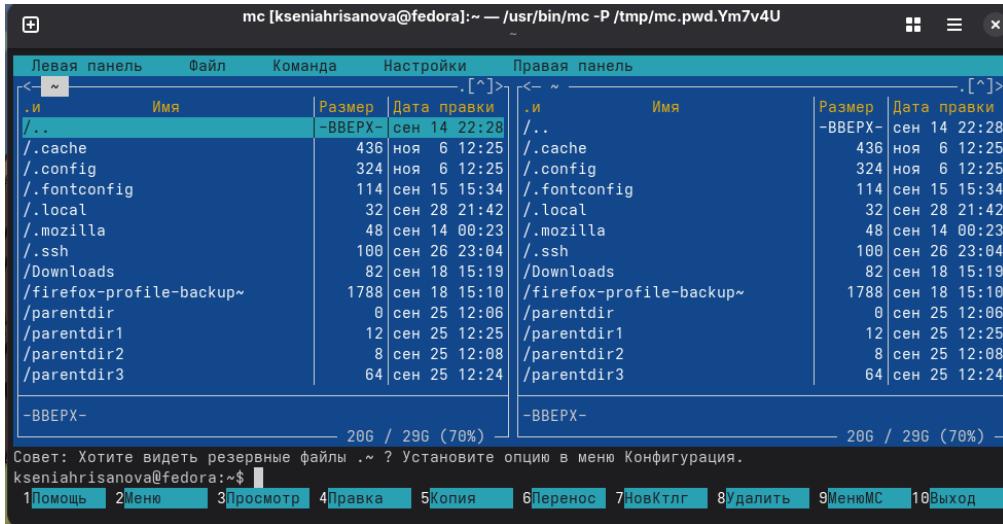


Рис. 2: Интерфейс Midnight Commander

Перехожу в созданный каталог в предыдущей лабораторной работе (рис. 3).

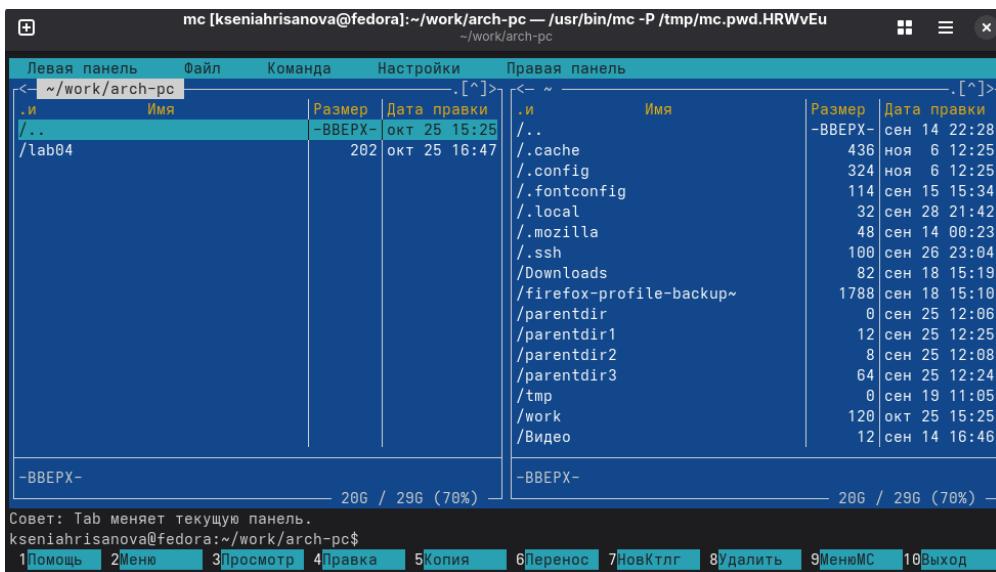


Рис. 3: Открытый каталог arch-pc

С помощью функциональной клавиши, я создаю подкаталог lab05, в котором буду работать (рис. 4).

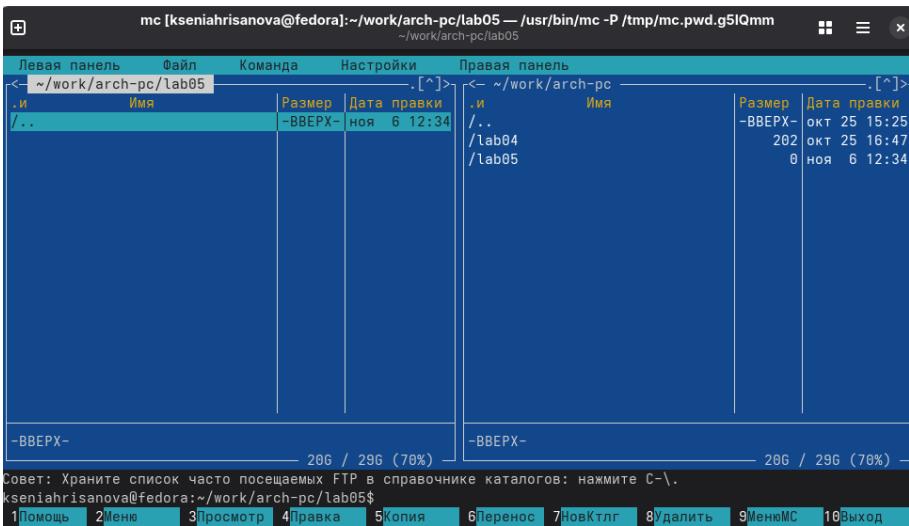


Рис. 4: Создание рабочего подкатализа

В строке ввода ввожу команду touch и создаю файл (рис. 5).

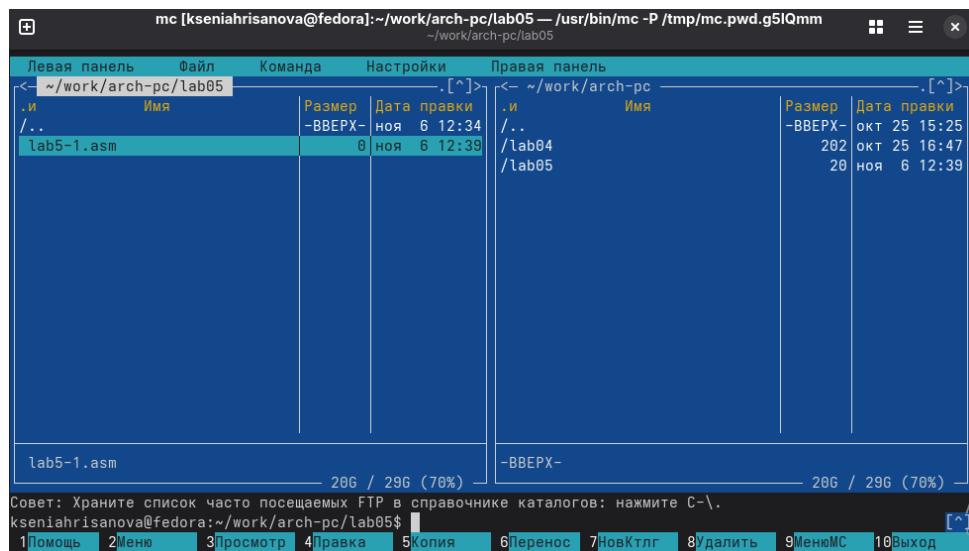


Рис. 5: Создание файла в Midnight Commander

4.2 Работа в NASM

С помощью F4 открываю только что созданный файл и вношу код с листинга (рис. 6).

```

Lab5-1.asm [-M-0] 13 L:[ 1+12 13/ 25 ] *(167 / 331b) 0010 0x00A
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
...

```

Рис. 6: Редактирование файла в Midnight Commander

Проверяю сохраненные изменения с помощью клавиши F3 (рис. 7).

```

mc [kseniahrisanova@fedora]:~/work/arch-pc/lab05 — /usr/bin/mc -P /tmp/mc.pwd.g5IQmm
/home/kseniahrisanova/work/arch-pc/lab05/lab5-1.asm
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
...

```

Рис. 7: Проверка сохранения сделанных изменений

Транслирую и компоную измененный файл, запускаю (рис. 8).

```

kseniahrisanova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Хрисанова Ксения Олеговна
kseniahrisanova@fedora:~/work/arch-pc/lab05$

```

Рис. 8: Трансляция, компоновка и последующий запуск программы

4.3 Подключение внешнего файла

Скачанный с ТУИС файл сохраняю в общую папку на своем компьютере, на виртуальной машине в интерфейсе Midnight Commander перехожу в директорию общей папки, копирую файл в рабочий подкаталог. (рис. 9).

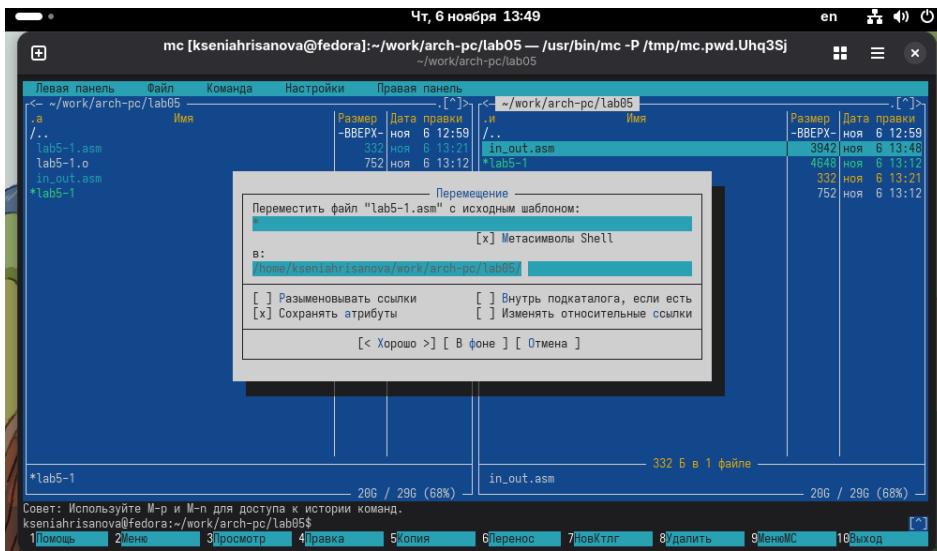


Рис. 9: Копирование файла в рабочий каталог

В копии файла подключаю подпрограмму из подключенного файла (рис. 10).

```
mc [kseniahrisanova@fedora:~/work/arch-pc/lab05 — /usr/bin/mc -P /tmp/mc.pwd.bbytuo
~/work/arch-pc/lab05]
GNU nano 8.3
;/home/kseniahrisanova/work/arch-pc/lab05/lab5-2.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры

%include "in_out.asm"      ; подключение внешнего файла с подпрограммами

SECTION .data
msg db 'Введите строку:', 0h ; сообщение для вывода

SECTION .bss
buf1 resb 80                ; буфер размером 80 байт

SECTION .text
GLOBAL _start

_start:
    mov eax, msg          ; адрес сообщения в EAX
    call sprintLF          ; вызвать подпрограмму вывода строки

    mov ecx, buf1          ; адрес буфера в ECX
    mov edx, 80              ; длина буфера
    call sread               ; вызвать подпрограмму ввода строки

    call quit                ; завершение программы
```

Рис. 10: Изменение программы

Транслирую, компоную и запускаю программу с подключенным файлом (рис. 11).

```
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ./lab5-1
Введите строку:
Хрисанова Ксения Олеговна
kseniahrisanova@fedora:~/work/arch-pc/lab05$
```

Рис. 11: Запуск измененной программы

Редактирую файл и заменяю в нем подпрограмму `sprintLF` на `sprint`. Разница подпрограмм в том, что вторая вызывает ввод на той же строке (рис. 12).

4.4 Задание для самостоятельной работы

```
kseniahrisanova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-2 lab5-2.o
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Хрисанова Ксения Олеговна
kseniahrisanova@fedora:~/work/arch-pc/lab05$
```

Рис. 12: Запуск изменной программы с другой подпрограммой

Транслирую, компоную и запускаю свою программу (рис. 13).

```
kseniahrisanova@fedora:~/work/arch-pc/lab05$ nasm -f elf lab5-3.asm
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-3 lab5-3.o
kseniahrisanova@fedora:~/work/arch-pc/lab05$ ./lab5-3
Введите строку:
Хрисанова Ксения Олеговна
Хрисанова Ксения Олеговна

kseniahrisanova@fedora:~/work/arch-pc/lab05$ █
```

Рис. 13: Запуск своей программы

5 Выводы

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

Список литературы

1. Пример выполнения лабораторной работы
2. Курс на ТУИС
3. Лабораторная работа №5