

Отчёт по лабораторной работе №5

Архитектура компьютера

Агапова Анна Антоновна

Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Выводы	11
4 Список литературы	12

Список иллюстраций

2.1	Открытие Midnight Commander	6
2.2	Переход в каталог	6
2.3	Создание папки.....	7
2.4	Переход в каталог	7
2.5	Создание файла.....	7
2.6	Проверка создания файла.....	7
2.7	Ввод текста программы.....	7
2.8	Проверка наличия текста программы	8
2.9	Транслирование текста, поверка работы	8
2.10	Копирование файла	8
2.11	Проверка копирования файла	9
2.12	Создание копии файла.....	9
2.13	Исправление текста программы.....	9
2.14	Проверка работы.....	9
2.15	Проверка работы.....	10
2.16	Проверка работы.....	10

Список таблиц

1 Цель работы

Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

2 Выполнение лабораторной работы

1. Открываю Midnight Commander с помощью команды mc. (рис. 2.1).

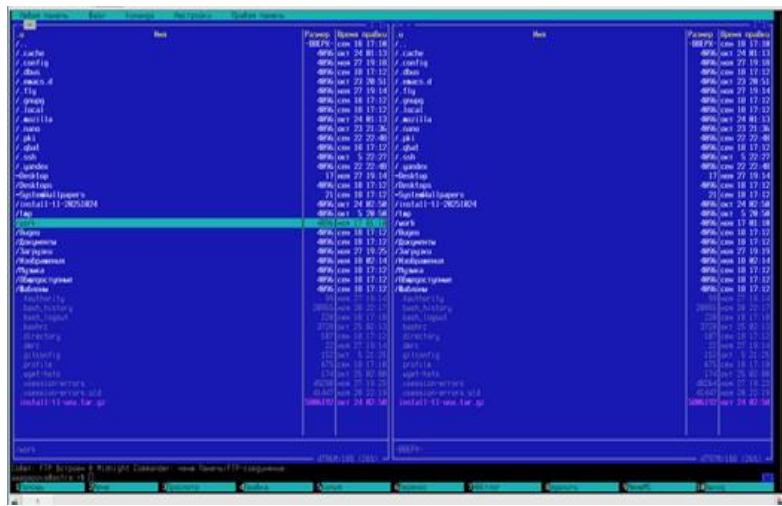


Рисунок 2.1: Открытие Midnight Commander

2. Пользуясь клавишами клавиатуры перехожу в каталог `~/work/arch-pc` созданный при выполнении лабораторной работы №4. (рис. 2.2).

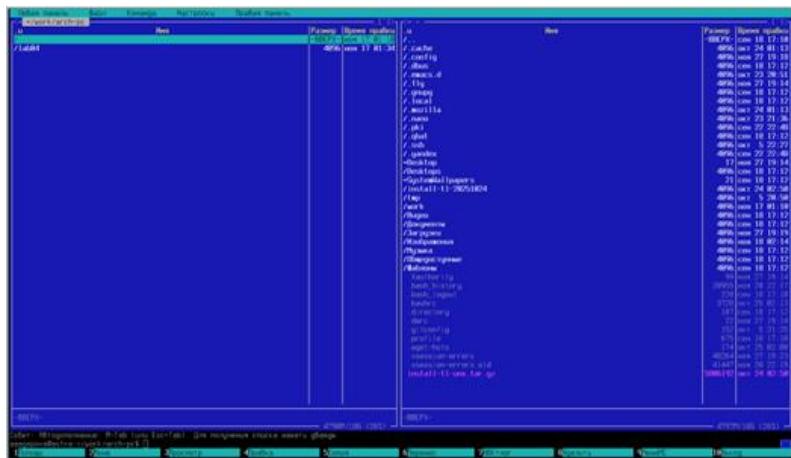


Рисунок 2.2: Переход в каталог

3. С помощью функциональной клавиши F7 создаю папку lab05 (рис. 2.3) и перехожу в созданный каталог. (рис. 2.4)

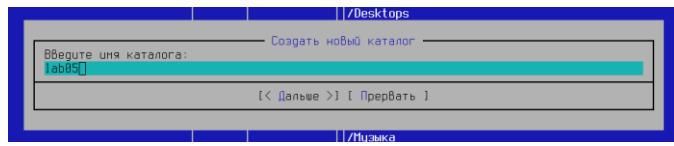


Рисунок 2.3: Создание папки

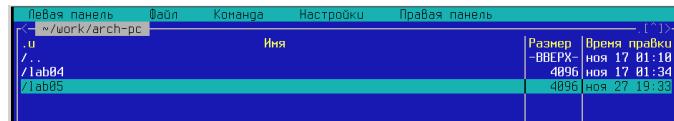


Рисунок 2.4: Переход в каталог

4. Пользуясь строкой ввода и командой touch создаю файл lab5-1.asm. (рис. 2.5)

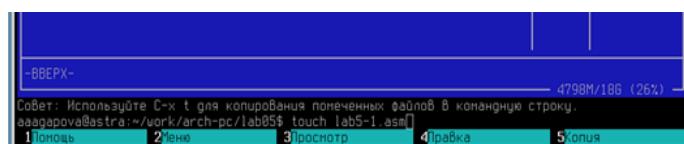


Рисунок 2.5: Создание файла

5. Проверю, что файл lab5-1.asm создался. (рис. 2.6).

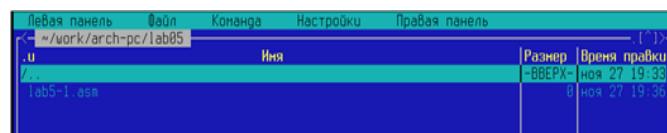


Рисунок 2.6: Проверка создания файла

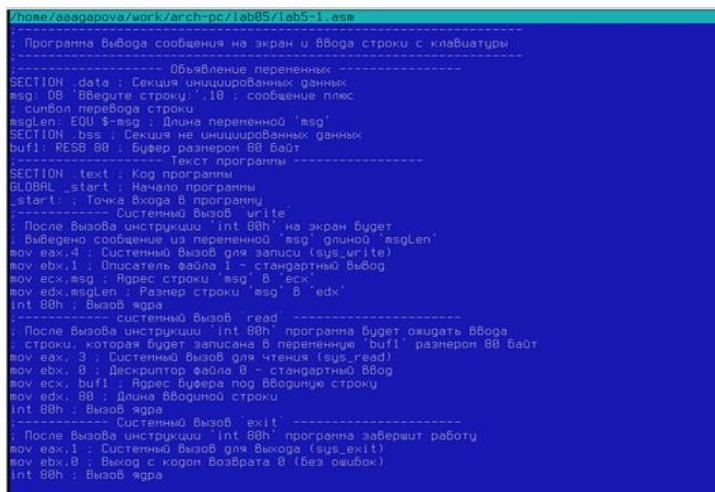
6. С помощью функциональной клавиши F4 открываю файл lab5-1.asm для редактирования во встроенным редакторе. Ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл (рис. 2.7)

```
ВИД файла 2:7:4   файл: /home/aaaararova/work/arch-pc/lab85/lab5-1.asm

; Программа выбора сообщения на экран и блюра строки с клавиатурой
;----- Объявление переменных -----
SECTION data ; Секция иммодифицируемых данных
msg:    .string "Hello $-word. Длина переменной $00"
SECTION bss ; Секция не инициализированных данных
buff:   .resw 80 ; Выдел размером 80 байт
;----- Определение программного кода -----
SECTION code ; Код программы
_start: start ; Начало программы
        ; Точка входа в программу
        ;----- Вводим строку из консоли -----
        ; После выполнения инструкции 'int 80h' на экран будет
        ; выведено сообщение из переменной msg и строка ввода
        ; введена в буфер блоффа buff
        ;----- Выводим строку из консоли -----
        mov dx, msg ; Печат строку 'Hello $-word'
        mov ah, 09h ; Режим строки msg в 'еск' режиме
        int 21h ; Вызов строки msg в 'еск' режиме
        ;----- Вводим системный файл read -----
        ; После выполнения инструкции 'int 80h' программа будет ожидать ввода
        ; строки, которая будет записана в переменную buff1 размером 80 байт
        ; введена в буфер блоффа buff1
        ;----- Выводим строку из консоли -----
        ; После выполнения инструкции 'int 80h' программа заберет работу
        ; от пользователя и сгенерирует ошибку (sys_exit)
        ; введена в буфер блоффа buff1
        int 21h ; Вызов строки msg в 'еск' режиме
        ;----- Вызов системного файла exit -----
        ; После выполнения инструкции 'int 80h' программа заберет работу
        ; от пользователя и сгенерирует ошибку (sys_exit)
        ; введена в буфер блоффа buff1
        int 80h ; Вызов sys_exit
```

Рисунок 2.7: Ввод текста программы

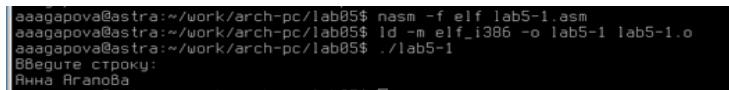
7. С помощью функциональной клавиши F3 открываю файл lab5-1.asm для просмотра. Убеждаюсь, что файл содержит текст программы. (рис. 2.8).



```
/home/aaagapova/work/arch-pc/lab05/lab5-1.asm
; Программа вывода сообщения на экран и ввода строки с клавиатуры
----- Объявление переменных -----
SECTION .data : Секция инициализированных данных
msg: DB 'Выведите строку:',10 ; сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной 'msg'
SECTION .bss : Секция не инициализированных данных
buf: RESB 80 ; Буфер размером 80 байт
----- Текст программы -----
SECTION .text : Код программы
GLOBAL _start : Наименование точки входа
_start: Точка входа в программу
        ;----- Системный вызов write -----
        ; После вывода инструкции 'int 80h' на экран будет
        ; выведено сообщение из переменной 'msg' полно '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 'еск'
        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 'buf' размером 80 байт
        mov eax,3 ; Системный вызов для чтения (sys_read)
        mov ebx,0 ; Дескриптор файла 0 - стандартный ввод
        mov ecx,buf ; 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
```

Рисунок 2.8: Проверка наличия текста программы

8. Отранслирую текст программы lab5-1.asm в объектный файл. Выполняю компоновку объектного файла и запускаю получившийся исполняемый файл (рис. 2.9).



```
aaagapova@astra:~/work/arch-pc/lab05$ nasm -f elf lab5-1.asm
aaagapova@astra:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-1 lab5-1.o
aaagapova@astra:~/work/arch-pc/lab05$ ./lab5-1
Выведите строку:
Анна Агапова
```

Рисунок 2.9: Транслирование текста, проверка работы

9. Скачиваю файл in_out.asm со страницы курса в ТУИС. В одной из панелей тс открываю каталог с файлом lab5-1.asm. В другой панели каталог со скачанным файлом in_out.asm. Скопирую файл in_out.asm в каталог с файлом lab5-1.asm с помощью функциональной клавиши F5. (рис. 2.10).

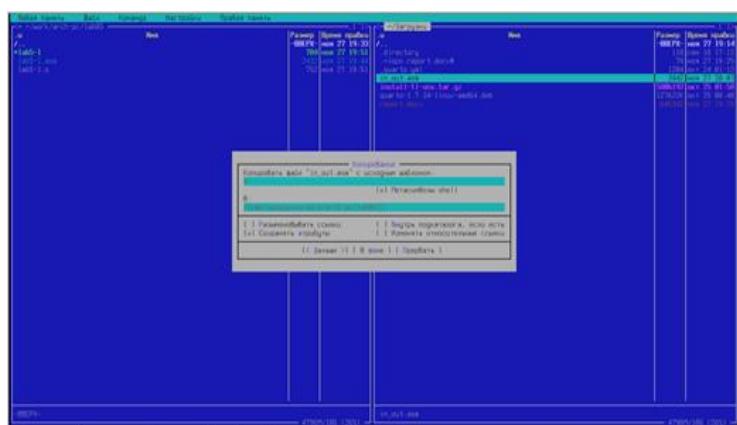


Рисунок 2.10: Копирование файла

10. Проверю, что файл `in_out.asm` скопировался в каталог с файлом `lab5-1.asm`. (рис. 2.11).

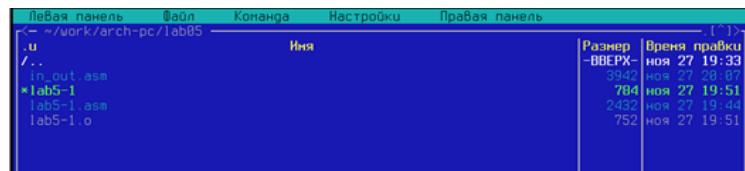


Рисунок 2.11: Проверка копирования файла

11. С помощью функциональной клавиши F6 создаю копию файла `lab5-1.asm` с именем `lab5-2.asm`. (рис. 2.12).

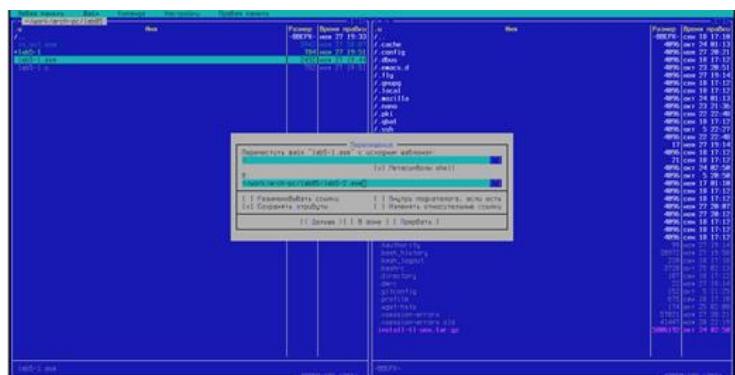


Рисунок 2.12: Создание копии файла

12. Исправляю текст программы в файле `lab5-2.asm` с использованием подпрограмм из внешнего файла `in_out.asm`, используя подпрограммы `sprintLF`, `sread` и `quit` в соответствии с листингом 5.2. (рис. 2.13).

```
/home/aaagapova/work/arch-pc/lab05/lab5-2.asm [---] 8 L:[ 1+ 8 / 18 ]*(8 /1255) 0059 0x036
Программа вывода сообщения на экран и ввода строки с клавиатуры

%include 'in_out.asm' ; подключение внешнего файла
SECTION data: ; Секция инициализации данных
msg: DB 'Введите строку:',0h ; сообщение
SECTION nos: ; Секция не инициализации данных
buff: RESB 80 ; Буфер размером 80 байт
SECTION text: ; Код программы
GLOBAL _start; ; Начало программы
_start: ; Точка входа в программу
    mov eax, msg ; запись адреса выведенного сообщения в EBX
    call sprintLF ; Вызов подпрограммы печати сообщения
    mov eax, buff ; запись адреса переменной в EBX
    mov edx, 80 ; запись флага ввода сообщения в EBX
    call sread ; Вызов подпрограммы ввода сообщения
    call quit ; Вызов подпрограммы завершения
```

Рисунок 2.13: Исправление текста программы

13. Проверим работу исправленного текста программы. (рис. 2.14).

```
aaagapova@astra:~/work/arch-pc/lab05$ nasm -f elf lab5-2.asm
aaagapova@astra:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab2-1 lab5-2.o
aaagapova@astra:~/work/arch-pc/lab05$ ./lab5-2
Введите строку:
Аня
```

Рисунок 2.14: Проверка работы

14. Создаю копию файла lab5-1.asm. Внесу изменения в программу (без использования внешнего файла in_out.asm) и проверяю его работу. (рис. 2.15).

```
aaagapova@astra:~/work/arch-pc/lab05$ nasm -f elf lab5-11.asm
aaagapova@astra:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-11 lab5-11.o
aaagapova@astra:~/work/arch-pc/lab05$ ./lab5-11
Введите строку:
Agapova
Вы ввели:
Agapova
```

Рисунок 2.15: Проверка работы

15. Создаю копию файла lab5-2.asm. Исправляю текст программы с использование подпрограмм из внешнего файла in_out.asm и проверяю его работу. (рис. 2.16)

```
aaagapova@astra:~/work/arch-pc/lab05$ nasm -f elf lab5-22.asm
aaagapova@astra:~/work/arch-pc/lab05$ ld -m elf_i386 -o lab5-22 lab5-22.o
aaagapova@astra:~/work/arch-pc/lab05$ ./lab5-22
Введите строку:
Agapova
Agapova
```

Рисунок 2.16: Проверка работы

3 Выводы

В ходе лабораторной работы мною были приобретены практические навыки работы в Midnight Commander и освоение инструкций языка ассемблера mov и int.

4 Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 9781491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.
10. Куляс О. Л., Никитин К. А. Курс программирования на ASSEMBLER. — М. : Солон-Пресс, 2017.
11. Новожилов О. П. Архитектура ЭВМ и систем. — М. : Юрайт, 2016.
12. Расширенный ассемблер: NASM. — 2021. — URL: <https://www.opennet.ru/docs/RUS/nasm/>.
13. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — 2-е изд. — БХВПетербург, 2010. — 656 с. — ISBN 978-5-94157-538-1.
14. Столяров А. Программирование на языке ассемблера NASM для ОС Unix. — 2-е изд. — М. : МАКС Пресс, 2011. — URL: http://www.stolyarov.info/books/asm_unix.
15. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 874 с. — (Классика Computer Science).
16. Таненбаум Э., Бос Х. Современные операционные системы. — 4-е изд. — СПб. : Питер, 2015. — 1120 с. — (Классика Computer Science).