

Лабораторная работа №5

Дисциплина: архитектура компьютера

-04-24

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.0.1	Подключение внешнего файла in_out.asm	9
3.0.2	Задание для самостоятельной работы	10
4	Выводы	12

Список иллюстраций

3.1	Создание папки	7
3.2	Редактирование программы	7
3.3	Открываю файл для просмотра	8
3.4	Транслирую текст программы в объектный файл	8
3.5	Исполняемый файл	8
3.6	Ввод текста для программы	9
3.7	Переместила файл в нужный каталог	9
3.8	Копирую файл	9
3.9	Проверка работы файла	10
3.10	Работа программы после внесения изменений	10
3.11	Задания 1,2 для самостоятельной работы	10
3.12	Задания 3,4 для самостоятельной работы	11

Список таблиц

1 Цель работы

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

2 Задание

1. Работа в Midnight Commander.
2. Подключение внешнего файла in_out.asm.
3. Задание для самостоятельной работы.

3 Выполнение лабораторной работы

*Открываю Midniht Commander. В каталоге созданном при выполнении лаб. работы №4 и там создаю папку lab05 (рис. 3.1).

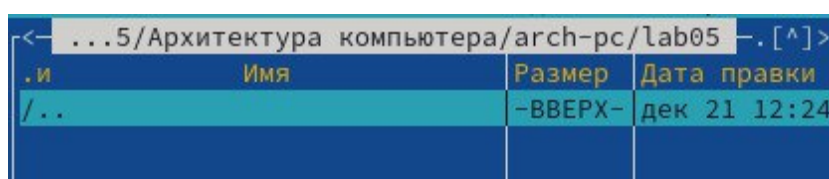


Рис. 3.1: Создание папки

*Создаю файл lab05-1.asm; Затем я открываю файл для редактирования во встроенном редакторе и ввожу текст программы из листинга 5.1, сохраняю изменения и закрываю файл (рис. 3.2).

```
lab05-1.asm      [-M--] 24 L:[ 1+22 23/ 26] *(1222/1371b) 0010 0x000f
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку:',10 ; Сообщение плюс
; символ перевода строки
msgLen: EQU $-msg ; Длина переменной.
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'
mov eax,4 ; Системный вызов для записи
mov ebx,1 ; Описатель файла 1 - стандартный вывод
mov ecx,msg ; Адрес строки
mov edx,msgLen ; Размер строки
int 80h ; Вызов ядра
; системный вызов 'read'
mov eax, 3 ; Системный вызов для чтения
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'
mov eax,1 ; Системный вызов для выхода
mov ebx,0 ; Выход с кодом возврата 0.
int 80h ; Вызов ядра
```

Рис. 3.2: Редактирование программы

*С помощью функциональной клавиши F3 открываю мой файл для просмотра.
Убеждаюсь, что текст программы правильный. (рис. 3.3).

```

/home/aesandan/work/study/2024-2025/Архи-ура компьютера/arch-pc/lab05/lab05-1.asm
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
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)

```

Рис. 3.3: Открываю файл для просмотра

*Транслирую текст программы lab05-1.asm в объектный файл (рис. 3.4).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-1.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-1 lab05-1.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$

```

Рис. 3.4: Транслирую текст программы в объектный файл

*Созданный мной исполняемый файл: (рис. 3.5).

*lab05-1	8748	дек 21 13:49
lab05-1.asm	2026	дек 21 13:46
lab05-1.o	752	дек 21 13:48

Рис. 3.5: Исполняемый файл

*После запуска мой исполняемый файл работает, он ожидает ввода с клавиатуры. Я ввожу своё ФИО(рис. 3.6).


```

aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-1
Введите строку:
Sandan Alina
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$

```

Рис. 3.6: Ввод текста для программы

3.0.1 Подключение внешнего файла in_out.asm

*Скачиваю нужный файл для выполнения лаб. работы из туиса, обращаю внимание на то, что скачанный файл должен находиться в том же каталоге, что и файл с программой (рис. 3.7).

```

./..
in_out.asm
*lab05-1
lab05-1.o

```

Рис. 3.7: Переместила файл в нужный каталог

*При помощи f6 я создаю копию файла lab05-1.asm с именем lab05-2.asm (рис. 3.8).

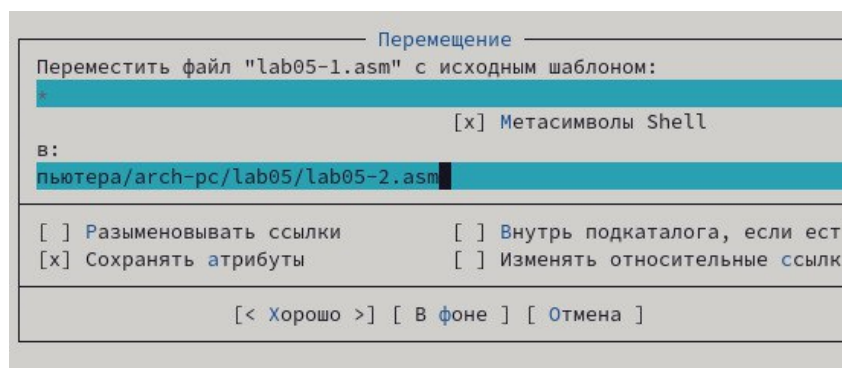


Рис. 3.8: Копирую файл

*Я исправляю текст программы в соответствии с листингом 5.2, затем создаю исполняемый файл и проверяю его работу(похожие действия мы уже проделывали в предыдущих подпунктах) (рис. 3.9).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-2.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-2 lab05-2.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-2
Введите строку:

```

Рис. 3.9: Проверка работы файла

*В файле lab05-2.asm мне нужно внести изменения в программу. После всех изменений я проверяю исполняемый файл на работу. Отличие в том, что теперь ввод с клавиатуры происходит не на следующей строчке, а на этой же. (рис. 3.10).

```

aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-2.asm
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-2 lab05-2.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-2
Введите строку:

```

Рис. 3.10: Работа программы после внесения изменений

3.0.2 Задание для самостоятельной работы

*Создаю копию файла lab05-1.asm и вношу изменения в программу, так чтоб она работала по алгоритму и вывормдила введеную строку на экран. Получаю исполняемый файл и проверяю его работу (рис. 3.11).

```

lab05-3.asm: warning: label 'stone' on a line without a colon might be an error [-Wlabel-operand]
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-3 lab05-3.o
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-3
Введите строку:
Sandan
Sandan
aesandan@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$

```

Рис. 3.11: Задания 1,2 для самостоятельной работы

*Создаю копию файла lab05-2.asm и исправляю текст с использованием программ из внешнего файла так, чтоб она опять таки выводила введеную команду на экран. Остается проверить работу программы: (рис. 3.12).

```
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ nasm -f elf lab05-3.asm
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ld -m elf_i386 -o lab05-3 lab05-3.o
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$ ./lab05-3
Введите строку: Sandan
Sandan
aesandan@fedora: ~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab05$
```

Рис. 3.12: Задания 3,4 для самостоятельной работы

4 Выводы

Я приобрела практические навыки работы в Midnight Commander. Освоила инструкции языка ассемблера mov и in. Я считаю, что приобретенные навыки во многом помогут мне при программировании.