

Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютеров

Пестова Ева Константиновна

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	1
4	Выводы.....	9

1 Цель работы

Целью данной лабораторной работы является приобретение практических навыков работы в Midnight Commander, освоение инструкций языка ассемблера mov и int.

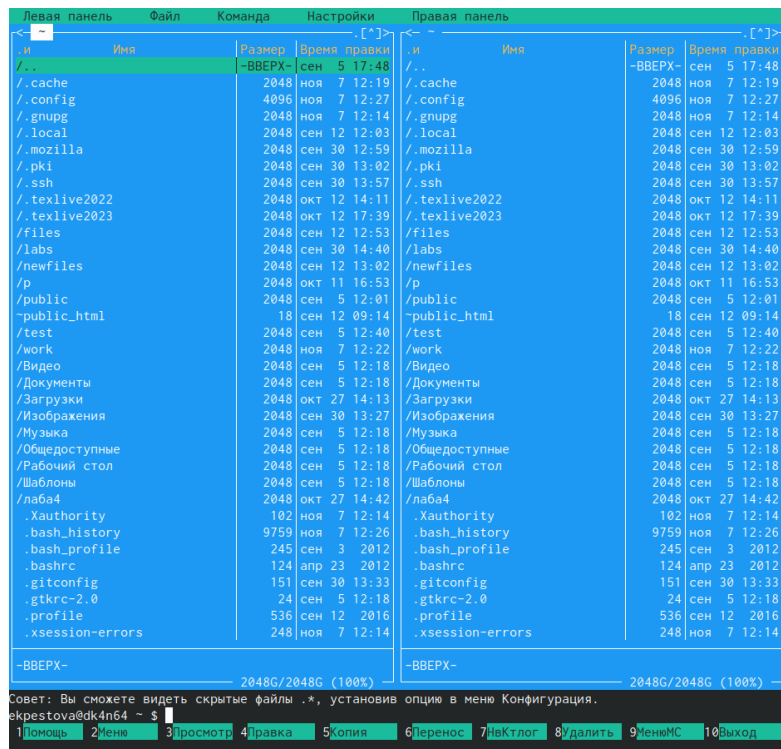
2 Задание

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

3 Выполнение лабораторной работы

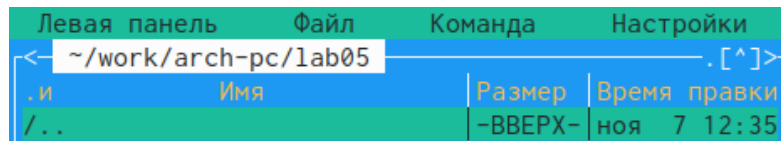
1. Основы работы с mc

Открываю Midnight Commander, введя в терминал mc (рис. [??]).



Midnight Commander

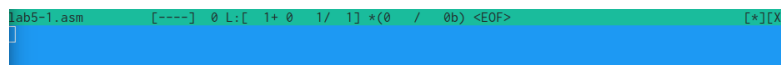
Перехожу в каталог `~/work/study/2022-2023/Архитектура Компьютера/arch-рс`, С помощью функциональной клавиши F7 создаю каталог `lab05` и перехожу в созданный каталог (рис. [??]).



Создание каталога

2. Структура программы на языке ассемблера NASM

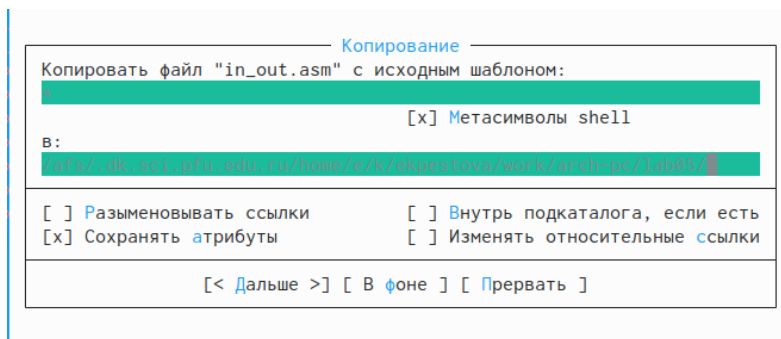
Создаю файл, в котором буду работать и с помощью функциональной клавиши F4 открываю созданный файл для редактирования (рис. [??]).



Создание файла

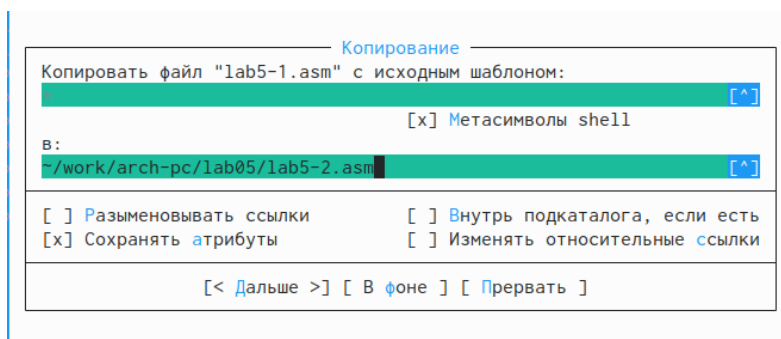
Ввожу в файл код программы для запроса строки у пользователя. Далее выхожу из файла, сохраняя изменения (рис. [??]).

С помощью функциональной клавиши F5 копирую файл in_out.asm из каталога Загрузки в созданный каталог lab05 (рис. [??]).



Копирование файла

С помощью функциональной клавиши F5 копирую файл lab5-1 в тот же каталог, но с другим именем, для этого в появившемся окне mc прописываю имя для копии файла (рис. [??]).



Копирование файла

Изменяю содержимое файла lab5-2.asm (рис. [??]), чтобы в программе использовались подпрограммы из внешнего файла in_out.asm.

```
GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/e/k/ekpestova/
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных 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 sprintf ; 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
```

Редактирование файла

Транслирую текст программы файла в объектный файл командой `nasm -f elf lab5-2.asm`. Создался объектный файл `lab5-2.o`. Выполняю компоновку объектного файла с помощью команды `ld -m elf_i386 -o lab5-2 lab5-2.o`. Создался исполняемый файл `lab5-2`. Запускаю исполняемый файл (рис. [??]).

```
nasm -f elf lab5-2.asm
ld -m elf_i386 -o lab5-2 lab5-2.o
```

Исполнение файла

Открываю файл `lab5-2.asm` для редактирования в nano функциональной клавишей F4. Изменяю в нем подпрограмму `sprintf` на `sprint`. Сохраняю изменения и открываю файл для просмотра, чтобы проверить сохранение действий (рис. [??]).

```
lab5-2.asm [B---] 41 L: [ 1+13 14/ 15] *(961 / 962b) 00
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция иницированных данных
msg: DB 'Введите строку: ',0h ; сообщение
SECTION .bss ; Секция не иницированных 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
```

Отредактированный файл

Снова транслирую файл, выполняю компоновку созданного объектного файла, запускаю новый исполняемый файл (рис. [??]).

```

ekpestova@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2.asm
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-2 lab5-2.o
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-2-2
Введите строку: Пестова Ева Константиновна

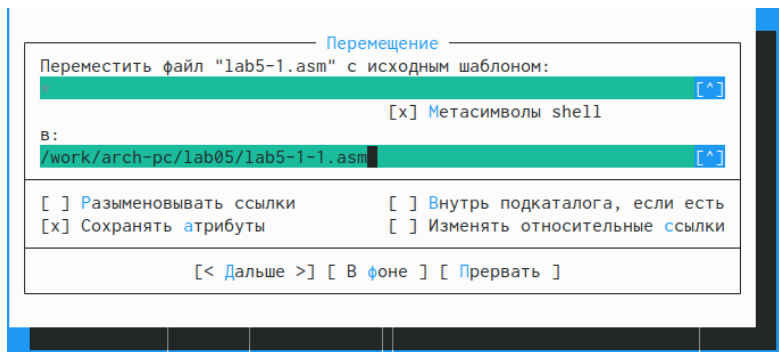
```

Исполнение файла

Разница между первым исполняемым файлом lab6-2 и вторым lab6-2-2 в том, что запуск первого запрашивает ввод с новой строки, а программа, которая исполняется при запуске второго, запрашивает ввод без переноса на новую строку, потому что в этом заключается различие между подпрограммами `sprintLF` и `sprint`.

4. Выполнение заданий для самостоятельной работы

1.1 Создаю копию файла `lab5-1.asm` с именем `lab5-1-1.asm` с помощью функциональной клавиши F5 (рис. [??]).



Копирование файла

1.2 С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. [??]).

```

lab5-1-1.asm          [----]  9 L: [ 1+39  40/ 41] *(2495/2496b) 0010 0x00A
;~~~~~
; Программа вывода сообщения на экран и ввода строки с клавиатуры
;~~~~~
;~~~~~ Объявление переменных ~~~~~
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
;~~~~~ 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 ; Файловый дескриптор 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,4 ; Системный вызов для выхода (sys_exit)
mov ebx,1 ; Выход с кодом возврата 0 (без ошибок)
mov ecx,buf1 ; Вызов ядра
mov edx,buf1 ;
int 80h ;
mov eax,1 ;
mov ebx,0 ;
int 80h ;

```

Редактирование файла

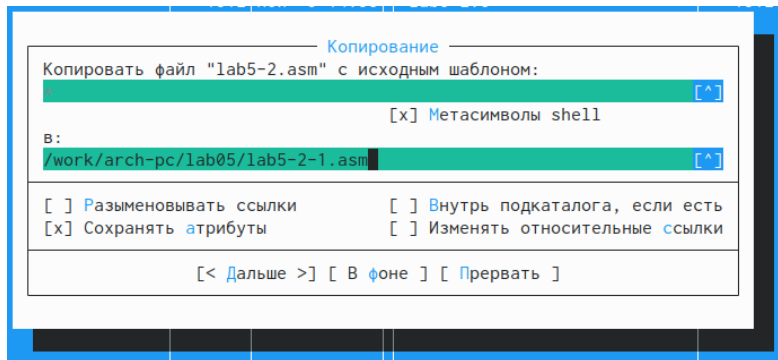
Код программы из пункта 1: SECTION .data ; Секция инициализированных данных msg: DB 'Введите строку:',10 msgLen: EQU \$-msg ; Длина переменной 'msg' SECTION .bss ; Секция не инициализированных данных buf1: RESB 80 ; Буфер размером 80 байт SECTION .text ; Код программы GLOBAL _start ; Начало программы _start: ; Точка входа в программу 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 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 mov eax,4 ; Системный вызов для записи (sys_write) mov ebx,1 ; Описатель файла '1' - стандартный вывод mov ecx,buf1 ; Адрес строки buf1 в ecx mov edx,buf1 ; Размер строки buf1 int 80h ; Вызов ядра mov eax,1 ; Системный вызов для выхода (sys_exit) mov ebx,0 ; Выход с кодом возврата 0 (без ошибок) int 80h ; Вызов ядра

2.1 Создаю объектный файл lab5-1-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-1-1, запускаю полученный исполняемый файл. Программа запрашивает ввод, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. [??]).

```
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-1-1.asm
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-1-1 lab5-1-1.o
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-1-1
Введите строку:
Пестова Ева Константиновна
Пестова Ева Константиновна
```

Исполнение файла

3.1 Создаю копию файла lab5-2.asm с именем lab5-2-1.asm с помощью функциональной клавиши F5 (рис. [??]).



Копирование файла

3.2 С помощью функциональной клавиши F4 открываю созданный файл для редактирования. Изменяю программу так, чтобы кроме вывода приглашения и запроса ввода, она выводила вводимую пользователем строку (рис. [??]).

```
GNU nano 7.2 /afs/.dk.sci.pfu.edu.ru/home/e/k/ekpestova/work/arch-pc/lab05/lab5-2-1.asm
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data ; Секция инициализированных данных
msg: DB 'Введите строку: ',0h ; сообщение
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
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
mov eax, 4 ;
mov ebx, 1 ;
mov ecx, buf1 ;
int 80h ;
call quit ;
```

Редактирование файла

Код программы из пункта 3: %include 'in_out.asm' SECTION .data ; Секция инициализированных данных msg: DB 'Введите строку:',0h ; сообщение 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 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 mov eax, 4 ; Системный вызов для записи (sys_write) mov ebx, 1 ; Описатель файла '1' - стандартный вывод mov ecx, buf1 ; Адрес строки buf1 в ecx int 80h ; Вызов ядра call quit ; вызов подпрограммы завершения

4.1 Создаю объектный файл lab5-2-1.o, отдаю его на обработку компоновщику, получаю исполняемый файл lab5-2-1, запускаю полученный исполняемый файл. Программа запрашивает ввод без переноса на новую строку, ввожу свои ФИО, далее программа выводит введенные мною данные (рис. [??]).

```
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5-2-1.asm
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 -o lab5-2-1 lab5-2-1.o
ekpestova@dk2n24 ~/work/arch-pc/lab05 $ ./lab5-2-1
Введите строку: Пестова Ева Константиновна
Пестова Ева Константиновна
```

Исполнение файла

4 Выводы

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