Отчёт по лабораторной работе №5

Дисциплина: архитектура компьютеров

Пестова Ева Константиновна

Содержание

# 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](data:application/octet-stream;base64,)

Midnight Commander

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

![Создание каталога](data:application/octet-stream;base64,)

Создание каталога

1. 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](data:application/octet-stream;base64,)

Создание файла

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

![Редактирование файла](data:application/octet-stream;base64,)

Редактирование файла

Транслирую текст программы файла в объектный файл командой nasm -f elf lab5-1.asm. Создался объектный файл lab5-1.o. Выполняю компоновку объектного файла с помощью команды ld -m elf\_i386 -o lab5-1 lab5-1.o . Создался исполняемый файл lab5-1. Запускаю исполняемый файл. Программа выводит строку “Введите строку:” и ждет ввода с клавиатуры, я ввожу свои ФИО, на этом программа заканчивает свою работу (рис. [??]).

![Компиляция и исполнение файла](data:application/octet-stream;base64,)

Компиляция и исполнение файла

1. Подключение внешнего файла

Скачиваю файл in\_out.asm со страницы курса в ТУИС. Он сохранился в каталог “Загрузки” (рис. [??]).

![Скачанный файл](data:application/octet-stream;base64,)

Скачанный файл

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

![Копирование файла](data:application/octet-stream;base64,)

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](data:application/octet-stream;base64,)

Копирование файла

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

![Редактирование файла](data:application/octet-stream;base64,)

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. [??]).

![Исполнение файла](data:application/octet-stream;base64,)

Исполнение файла

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

![Отредактированный файл](data:application/octet-stream;base64,)

Отредактированный файл

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

![Исполнение файла](data:application/octet-stream;base64,)

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.

1. 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](data:application/octet-stream;base64,)

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. [??]).

![Редактирование файла](data:application/octet-stream;base64,)

Редактирование файла

Код программы из пункта 1: SECTION .data ; Секция инициированных данных msg: DB ‘Введите строку:’,10 msgLen: EQU $-msg ; Длина переменной ‘msg’ 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,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. [??]).

![Исполнение файла](data:application/octet-stream;base64,)

Исполнение файла

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

![Копирование файла](data:application/octet-stream;base64,)

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. [??]).

![Редактирование файла](data:application/octet-stream;base64,)

Редактирование файла

Код программы из пункта 3: %include ‘in\_out.asm’ 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 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. [??]).

![Исполнение файла](data:application/octet-stream;base64,)

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.