Лабораторная работа №5

Отчет

Заболотная Кристина Александровна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Создать копию файла lab6-1.asm. Внести изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран.
2. Получить исполняемый файл и проверить его работу. На приглашение ввести строку, свою фамилию.
3. Создать копию файла lab6-2.asm. Исправить текст программы с использованием подпрограмм из внешнего файла in\_out.asm, так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. Подключаемый файл in\_out.asm должен лежать в том же каталоге, что и файл с программой, в которой он используется.
4. Создать исполняемый файл и проверить его работу.

# 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. Для активации оболочки Midnight Commander достаточно ввести в командной строке mc и нажать клавишу Enter. 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. Простейший диалог с пользователем требует наличия двух функций — вывода текста на экран и ввода текста с клавиатуры. Простейший способ вывести строку на экран — использовать системный вызов write. Этот системный вызов имеет номер 4, поэтому перед вызовом инструкции int необходимо поместить значение 4 в регистр eax. Первым аргументом write, помещаемым в регистр ebx, задаётся дескриптор файла. Для вывода на экран в качестве дескриптора файла нужно указать 1 (это означает «стандартный вывод», т. е. вывод на экран). Вторым аргументом задаётся адрес выводимой строки (помещаем его в регистр ecx, например, инструкцией mov ecx, msg). Строка может иметь любую длину. Последним аргументом (т.е. в регистре edx) должна задаваться максимальная длина выводимой строки. Для ввода строки с клавиатуры можно использовать аналогичный системный вызов read. Его аргументы – такие же, как у вызова write, только для «чтения» с клавиатуры используется файловый дескриптор 0 (стандартный ввод). Системный вызов exit является обязательным в конце любой программы на языке ассемблер. Для обозначения конца программы перед вызовом инструкции int 80h необходимо поместить в регистр еах значение 1, а в регистр ebx код завершения 0.

# 4 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. **¿fig:0051?**)

1. Откроем Midnight Commander. (рис. **¿fig:0051?**)

51.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. **¿fig:0052?**)

1. Создаем директорию. (рис. **¿fig:0052?**)

52.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. **¿fig:0053?**)

1. Создаем файл lab5-1.asm. (рис. **¿fig:0053?**)

53.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. **¿fig:0054?**)

1. Введим текст программы из листинга. (рис. **¿fig:0054?**)

54.png

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. **¿fig:0055?**)

1. Введим текст программы из листинга. (рис. **¿fig:0055?**)

55.png

# 5 Выводы

В ходе выполнения данной лабораторной работы были приобретены практические навыки работы в Midnight Commander. Былы освоены инструкции языка ассемблера mov и int.

# Список литературы