

Министерство образования и науки
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение
высшего профессионального образования
«Алтайский государственный технический университет
им.И.И.Ползунова»

Факультет информационных технологий

Сучкова Л.И.

Учебно-методическое пособие
по дисциплине
«Периферийные устройства»

Барнаул 2012

УДК 004.42

Сучкова, Л.И. Учебно-методическое пособие по дисциплине «Периферийные устройства»/ Л.И. Сучкова; АлтГТУ им. И.И. Ползунова. – Барнаул, АлтГТУ, 2012. – 34 с.

В учебно-методическом пособии по курсу «Периферийные устройства» приведены требования к содержанию лабораторных работ, методика их выполнения, индивидуальные задания, содержание отчетов. Пособие предназначено для студентов, обучающихся по направлению 231000 «Программная инженерия».

Содержание

Лабораторная работа № 1.....	4
Лабораторная работа № 2.....	10
Лабораторная работа № 3.....	16
Лабораторная работа № 4.....	23
Лабораторная работа № 5.....	25
Лабораторная работа № 6.....	32
Лабораторная работа № 7.....	32
Список литературы.....	34

Лабораторная работа № 1.

Цели и задачи работы: изучение макрогенерации, написание низкоуровневой программы с простейшим интерфейсом.

Теоретические сведения о работе приведены в [1,3,4,6,10], пример программы приведен в [10].

Задание к работе: Реализовать циклическую и условную макрогенерацию. Самостоятельно решить задачу в соответствии с индивидуальным вариантом.

Описание используемых средств для выполнения работы : язык программирования Assembler, операционная система Windows XP/7. Используется пакетная компиляция, компилятор tasm, компоновщик tlink.

Методика выполнения работы:

1. Изучить способы задания макроопределений, реализовать индивидуальное задание как макроопределение.
2. Написать и отладить программу с вводом-выводом информации без подключения макроопределения.
3. Подключить макроопределение. Отладить программу.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной приведены в [14]. Защита лабораторной, помимо теоретических вопросов, включает выполнение практического задания с написанием макроопределения, реализующего задание, предложенное преподавателем.

Индивидуальные задания:

Обычный уровень сложности:

1. Даны 3 квадратных матрицы различных размерностей. Для каждой из них заменить нулями все её четные элементы, расположенные на главной диагонали или выше неё. Подсчитать количество таких замен.
2. Даны 3 матрицы различных размерностей. Для каждой из них выяснить, сколько положительных элементов содержит матрица и найти их среднее арифметическое.
3. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти среднее арифметическое каждого столбца и полученные числа сложить.
4. Даны 2 квадратных матрицы различных размерностей. Для каждой из них получить еще две квадратные матрицы. В первую матрицу записать все элементы исходной выше главной диагонали, во вто-

рую – элементы, которые лежат ниже главной диагонали. Главную диагональ записать в обе новые матрицы.

5. Даны 3 матрицы различных размерностей. Для каждой из них получить массив b_1, \dots, b_n , где b_i – это значение первого по порядку положительного элемента i -ой строки матрицы (если таких элементов нет, то принять $b_i=0$).

6. Даны 3 квадратных матрицы различных размерностей. Для каждой из них построить последовательность b_1, \dots, b_n из нулей и единиц, в которой $b_i=0$ тогда, когда в i -ой строке матрицы есть хотя бы один равный нулю элемент

7. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти наименьшее из значений элементов столбца, который обладает наибольшей суммой элементов. Если таких столбцов несколько, то взять последний из них.

8. Даны 3 матрицы различных размерностей, все элементы которых различны. Для каждой из них в каждой строке выбрать элемент с наименьшим значением, затем среди этих чисел выбрать наибольшее.

9. Даны 3 матрицы различных размерностей. Для каждой матрицы найти массив, элементы которого представляют собой сумму наибольшего и наименьшего из значений элементов каждой строки матрицы.

10. Даны 3 квадратных матрицы разных размерностей. В каждой матрице найти число, равное сумме элементов строки, в которой расположен элемент с наименьшим значением. Полагается, что такой элемент единственный. Такие числа для всех матриц сложить.

11. Даны 3 матрицы различных размерностей. Для каждой матрицы найти последовательность b_1, \dots, b_n , где b_k – это число положительных элементов в k -ой строке.

12. Даны 3 матрицы различных размерностей. Для каждой матрицы найти последовательность b_1, \dots, b_n , где b_k – это сумма значений элементов, кратных 2, в k -ой строке.

13. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти сумму номеров строк, все элементы которых принадлежат интервалу от C до D и кратных 2. Полученные суммы для каждой матрицы сложить.

14. Даны 3 квадратных матрицы различных размерностей. Для каждой из них все элементы, равные наибольшему значению в матрице, заменить нулями. Число таких замен для каждой матрицы посчитать и вычислить общее количество замен во всех матрицах.

15. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти сумму номеров строк, элементы в каждой из которых образуют возрастающую последовательность.

16. Даны 3 квадратных матрицы различных размерностей. Для каждой из них в строках с ненулевым первым элементом найти произведение всех элементов и все кратные 4 произведения сложить.

17. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти сумму нечётных элементов в строках с элементом на главной диагонали, удовлетворяющим условию $A_{ii} > B$.

18. Даны 3 матрицы различных размерностей. Для каждой из них найти сумму последовательности b_1, \dots, b_n , состоящую из значений средних арифметических элементов строк.

19. Даны 3 матрицы различных размерностей. Для каждой из них найти сумму элементов столбцов, имеющих четные номера. Полученные значения сумм для каждой матрицы сложить и найти остаток от деления на 8.

20. Даны 3 квадратных матрицы различных размерностей. Для каждой из них в строках с нулевым элементом на главной диагонали найти сумму всех элементов. Проверить, если сумма может храниться в байте, то выдать ответ «Да», иначе «Нет».

21. Даны 3 квадратных матрицы различных размерностей. Для каждой из них найти сумму элементов строк, в которой расположены элементы с наименьшим значением. Предполагается, что такой элемент не единственный.

22. Даны 3 матрицы различных размерностей. Для каждой из них найти среднее арифметическое элементов каждого из столбцов, имеющих чётные номера.

23. Даны три числа размерности матриц. Сформировать 3 матрицы заданных размерностей по закону: $A_{ii}=0$ и $A_{ij}=(i*j)$ для $i,j=1, \dots, n$. Заданы также три числа-границы. Определить, сколько элементов, больших соответствующего числа-границы, содержит матрица.

24. Даны 3 квадратных матрицы различных размерностей. Для каждой из них получить b_1, \dots, b_n , где b_i -это наибольшее из значений элементов, находящихся в начале i -ой строки матрицы до элемента, принадлежащего главной диагонали включительно.

25. Даны 3 квадратных матрицы различных размерностей. Для каждой из них определить произведение элементов, индексы которых чётные.

26. Даны 3 матрицы различных размерностей. Для каждой из них определить среднее арифметическое элементов в нечётных столбцах.

Для каждого среднего арифметического найти целую часть от деления на 8.

27. Даны 3 квадратные матрицы различных размерностей. Для каждой из них определить сумму элементов последнего столбца и первой строки. Все полученные суммы сложить и разделить на число, равное минимальному из размерностей матриц.

28. Даны 3 квадратные матрицы различных размерностей. Для каждой из них определить произведение элементов каждой второй строки. Если в строке есть нули, то заменить их на единицы и сосчитать общее количество таких замен во всех матрицах.

29. Даны 3 матрицы различных размерностей. Для каждой из них определить минимальный элемент главной диагонали. Найти среднее арифметическое минимальных элементов. Если оно будет меньше 10, то выдать сообщение.

30. Даны 3 квадратные матрицы различных размерностей. Для каждой из них определить сумму элементов нечётных столбцов. Найти среднее арифметическое полученных чисел. Если оно будет больше 3, то выдать сообщение.

31. Даны 3 квадратные матрицы различных размерностей. Для каждой из них определить количество кратных 4 элементов в чётных строках. Для первой и третьей матрицы сложить полученные числа.

32. Даны 3 квадратные матрицы различных размерностей. Для каждой из них определить сумму элементов ниже главной диагонали. Если получится хотя бы одно четное число, то выдать сообщение.

33. Даны 3 матрицы различных размерностей. Для каждой из них вычислить максимальный элемент первой строки. Из полученных максимумов для всех матриц сформировать массив, в котором все элементы разделить на их номера.

34. Даны 3 квадратные матрицы различных размерностей. Для каждой из них вычислить сумму первого и последнего столбца. Все полученные суммы сложить и разделить на число, равное максимуму из размерностей.

35. Даны 4 матрицы различных размерностей. Для каждой из них определить сколько чётных чисел лежит выше главной диагонали.

36. Даны 3 матрицы различных размерностей. Для каждой из них сформировать массив, содержащий максимальные элементы по каждой из строк.

37. Даны 3 матрицы различных размерностей. Для каждой из них найти минимальный элемент. Если он больше заданного числа k , то в массив записать элементы главной диагонали матрицы, иначе массив заполнить нулями. Элементы этих массивов для всех матриц сложить.

38. Даны 3 матрицы различных размерностей. Для каждой из них в каждой строке найти убывающие последовательности, выдать их и их длину на экран.

39. Даны 3 квадратные матрицы различных размерностей. Каждую из них транспонировать и в её последней строке найти количество ненулевых элементов.

40. Даны 3 матрицы различных размерностей. Для каждой из них для всех элементов первой строки проанализировать, содержатся ли в матрице квадраты этих чисел.

41. Даны 4 матрицы различных размерностей. Проверить, какие из них можно перемножить, и если да, то умножить и в результирующей матрице найти максимальный элемент. Все полученные таким образом максимальные элементы сложить.

42. Даны 3 матрицы различных размерностей. Для каждой из них упорядочить строки матрицы по возрастанию элемента на главной диагонали.

43. Даны 3 матрицы различных размерностей и три массива. Для каждой из матриц проверить, содержит ли она все элементы соответствующего массива.

44. Даны 3 матрицы различных размерностей. Для каждой из них все чётные элементы матрицы записать в массив и определить, сколько элементов могут занимать один байт, а скольким одного байта не хватит.

45. Даны две матрицы, в которых хранятся координаты точек на плоскости. Для каждой матрицы определить, сколько точек лежит выше прямой, проходящей через точки с координатами (a, b) и (c, d). a, b, c, d задаются пользователем.

46. Даны 3 матрицы различных размерностей. Для каждой из них вывести на экран её элементы, кратные восьми. Если таких чисел нет, то вывести элемент в левом нижнем углу матрицы.

47. Даны 3 разреженные матрицы различных размерностей из 0 и 1. Для каждой из них сформировать список, содержащий номер строки и номер столбца для всех ненулевых элементов.

48. Даны три матрицы различных размерностей и 3 числа. Проверить, сколько элементов каждой матрицы по модулю превышает соответствующее матрице число (задается пользователем).

49. Даны три матрицы и три массива. Для каждой пары «матрица-массив» найти в матрице сумму элементов, принадлежащих отрезку от минимального до максимального значения соответствующего элемента массива.

50. Даны три матрицы и три пары чисел. В каждой матрице найти сумму элементов строки, где находится элемент, равный сумме соответствующей пары чисел.

51. Даны три матрицы. Для каждой из них все четные элементы выше главной диагонали записать в соответствующий массив и вывести на экран.

52. Даны три матрицы различных размерностей. В каждой из них найти среднее арифметическое минимального и максимального элементов.

53. Даны 3 квадратных матрицы. В каждом столбце найти произведение минимального и максимального по столбцу элемента.

Пониженный уровень сложности:

54. Даны 4 массива различных размерностей. Для каждого из них получить число, равное $n+1$, если $a_1 < a_2 < \dots < a_n$, иначе m , где m -длина возрастающей последовательности $a_1 < a_2 < \dots < a_m$ и $a_m > a_{m+1}$.

55. Даны 3 массива различных размерностей. Для каждого из них получить число l , равное длине убывающей последовательности b_1, b_2, b_3, \dots .

56. Даны 4 массива различных размерностей. Для каждого из них выяснить, сколько элементов с нечетными номерами являются четными числами.

57. Даны 3 массива различных размерностей. Для каждого из них выяснить, сколько элементов данного массива принадлежит заданному диапазону $[a; b]$.

58. Даны 4 массива различных размерностей. Для каждого из них определить, сколько элементов с четными номерами являются положительными числами.

59. Даны четыре пары натуральных чисел m и n . Для каждой из 4 матриц с заданными размерностями найти количество чисел, равное сумме размерностей для каждой матрицы.

60. Даны 4 массива различных размерностей. Для каждого из них определить, сколько в этом массиве возрастающих последовательностей.

61. Даны два массива A и B из n элементов. Если первый массив возрастает, а второй - убывает, то сформировать массив $ci = ai * bi$, иначе $ci = ai + bi$.

62. Даны два массива A и B из n элементов. Сформировать массив C по правилу $ci = \max(ai, bi)$. В массиве C найти сумму ненулевых элементов.

63. Даны 3 массива различных размерностей. Для каждого из них определить, сколько элементов в массиве превышает заданное число. Если такие элементы найдутся, то вычислить сумму этих элементов и разделить сумму на минимальную размерность массива.

64. Даны два массива - координаты x и y точек на плоскости. Определить номера точек, лежащих на осях координат, и вывести на экран координаты точки с наибольшим удалением от центра координат.

Лабораторная работа № 2.

Цели и задачи работы: изучение структуры обработчика прерывания, написание программы, позволяющей исследовать работу собственного обработчика в комплексе с системным обработчиком прерывания от клавиатуры.

Теоретические сведения о работе и пример программы приведены в [10].

Задание к работе: Написать обработчик прерываний от клавиатуры. Изучить и реализовать, если необходимо, отслеживание временных промежутков с помощью прерывания `1Ch`. Изучить перепрограммирование стандартного контроллера прерываний, реализацию маскирования линий `IRQ`. Самостоятельно решить задачу в соответствии с индивидуальным вариантом.

Описание используемых средств для выполнения работы : язык программирования `Assembler`, операционная система `Windows XP/7`. Используется пакетная компиляция, компилятор `tasm`, компоновщик `tlink`.

Методика выполнения работы:

1. Изучить алгоритм работы стандартного обработчика прерываний от клавиатуры.

2. Написать обработчик прерывания от клавиатуры, на первом этапе функционал своего обработчика не реализовывать или повторить тело обработчика по игнорированию клавиши «пробел» из [10].

3. Отладить программу. Убедиться в корректности работы обработчика.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной приведены в [10]. Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Индивидуальные задания:

1. Игнорировать нажатие всех клавиш, кроме клавиш основной клавиатуры (буквы, цифры), причем при нажатии ScrollLock восстанавливать функции клавиш.

2. Программно отключить обработку букв и цифр, а их ввод эмулировать нажатием Alt+F1 ... (Alt+F10) и набором ASCII кода при включенном CapsLock.

3. При нажатии клавиши CapsLock коды буквенных клавиш сдвигаются, т.е. в буфер помещается другой символ, в соответствии с количеством нажатий CapsLock+ASCII-код нажатого символа.

4. Написать обработчик прерываний от клавиатуры, функционирующий в двух режимах: 1) обычный, 2) ввод ASCII кодов букв в десятичном формате на основной клавиатуре. Переключение между режимами осуществляется при нажатии одной из клавиш F1- F5.

5. При нажатии ScrollLock осуществляется переключение режима работы клавиатуры: 1-е нажатие – отключение цифр; 2-е нажатие – включение цифр.

6. В обработчике прерывания от клавиатуры последовательно отключать алфавитно-цифровые клавиши по линейкам при нажатии Ctrl-ScrollLock (по 1 клавише при 1 нажатии). При нажатии Esc реакция на клавиши восстанавливается.

7. Промоделировать работу клавиатуры, которой управляют с помощью 4 клавиш. Нажатие Alt – переход в режим набора ASCII – кода символа, который набирается с помощью 0 и 1 в бинарном представлении. Отпускание Alt служит концом ввода кода символа. Клавиша Esc предназначена для восстановления обычного режима работы.

8. Написать обработчик прерываний от клавиатуры, который позволяет работать на частично неисправной клавиатуре – не работает линейка QWERTY...(отключить программно). Для набора букв этой линейки использовать Alt и цифровые клавиши. Ввод символов идет в режиме ввода десятичного ASCII –кода клавиши, пока Alt нажата. При ее отпускании формируются Scan и ASCII-коды предполагаемой клавиши.

9. На клавиатуре не работают клавиши ZXCVBNM, эмулировать их нажатием Ctrl + их номер в двоичном представлении, набранный на основной клавиатуре.

10. Написать обработчик прерываний от клавиатуры, при работе которого можно вводить ASCII-коды символов верхней строки (qwerty...) в двоичном коде с помощью клавиш ScrollLock и 0 и 1 на основной цифровой клавиатуре. Основной режим набора букв верхней линейки отключен.

11. При нажатии на клавиши от Q до R осуществляется циклический сдвиг выдаваемых кодов клавиш (при нажатии одной клавиши сдвиг осуществляется на 1)

12. С помощью клавиш Ctrl,0,1 эмулируется ввод скэн кода любой буквы. Перевод двоичного кода осуществляется в зависимости от состояния CapsLock.

13. Линейку клавиш qwer... заменить на клавишу “стрелка влево”, а ввод символов из этих линеек реализовать через нажатие Ctrl и цифры.

14. Линейку клавиш asd.. заменить на клавишу “стрелка вправо”, линейку zxc..– на клавишу “стрелка вниз”, а ввод символов реализовать через Shift и Alt и цифр.

15. Написать обработчик прерываний от клавиатуры, который после нажатия Ctrl(L)+Shift(L) формирует в буфере клавиатуры макроопределения `C → sprintf(); F → for() {};` `W → while() {}`. Повторное нажатие данной комбинации отменяет этот режим.

16. Предусмотреть обработчик прерываний от клавиатуры активизацию режима замены при нажатии LeftCtrl+Del. Дальнейшее нажатие любой клавиши в этом режиме заменяет соседние клавиши для нажатой на клавиши “влево” и “вправо”, а стрелки меняются на эти клавиши.

19. Написать обработчик прерываний от клавиатуры, который реализует зеркальное отображение линеек QWERTY... и ZXCVBN... клавиатуры относительно линейки ASDFG...

20. Клавиши Tab, LeftShift, BackSpace неисправны, поэтому эмулировать их нажатие через LeftCtrl+1,2,3 на цифровой клавиатуре соответственно.

21. Написать обработчик прерываний от клавиатуры, который при первом нажатии на клавиши алфавитно-цифровой клавиатуры выдает тот же символ, при втором нажатии – символ с ASCII – кодом, на 1 большим, в 3 раз – на 2 большим и т.д.

22. Написать обработчик прерываний от клавиатуры, выполняющий следующие действия: при нажатии Ctrl верхняя линейка (QWERTY...) начинает функционировать как цифры, при повторном нажатии Ctrl – продолжение нормального функционирования. Аналогичные действия осуществить со 2 линейкой при нажатии RShift.

23. На клавиатуре не работают цифровые клавиши. Эмулировать их нажатие с помощью 0 – Ctrl A, 1 – Ctrl B и т.д.

24. Написать обработчик прерываний от клавиатуры, который реагирует только один раз на нажатие алфавитно-цифровых клавиш. Для того, чтобы уже нажатые клавиши были доступны можно их вос-

становить нажатием: F1 - цифры, F2 – верхняя линейка, F3 - средняя линейка, F4 – нижняя линейка.

25. В обработчике прерываний от клавиатуры предусмотреть работу в двух режимах: обыкновенный ввод символов с консоли или ввод ASCII – кодов символов с цифровой клавиатуры. Переключение между режимами клавишами F3-F6.

26. Написать обработчик прерываний от клавиатуры, который при нажатии ScrollLock начинает функционировать следующим образом. Клавиша “пробел” заменяется на цифры от 0 до 9, при каждом следующем нажатии цифра увеличивается, после 9 опять становится равной 0.

27. Реализовать замену цифр 1-9 соответственно на буквы ABCDEFGHI. Такая замена осуществляется каждые 5 секунд из 10.

28. Гласным буквам латинского алфавита сопоставить строку текста, которая будет заноситься в буфер клавиатуры при нажатии этих букв.

29. В обработчике прерываний от клавиатуры реализовать замену букв на другие буквы. Информация о заменах хранится в таблице. Занесение в таблицу - нажатие F3, заменяемой буквы, новой буквы, нажатие F4. Очистка таблицы - нажатие F5.

30. Написать обработчик прерываний от клавиатуры, который каждые 10 с менял бы регистр нажимаемых клавиш (строчные на заглавные и наоборот) При нажатии на F1 восстанавливается нормальное состояние. При нажатии на F6 время задержки увеличивается на 0,5 с.

31. Написать обработчик прерываний от клавиатуры, который бы первые 5 с не реагировал на клавиши, вторые 5 с работал стандартно, и третьи 5 с реагировал только на пробел.

32. Каждые 10 с клавиши из верхнего ряда алфавитно-цифровой клавиатуры (Q . . .) заменяются на клавиши 2 ряда клавиатуры. При нажатии PrtScg восстанавливаются прежние буквы.

33. Написать обработчик прерываний от клавиатуры, осуществляющий циклический сдвиг цифр через промежуток t. При нажатии F1 t увеличивается в два раза, при нажатии Ctrl-Alt-Ins – уменьшается в 2 раза.

34. Написать обработчик прерываний от клавиатуры, который эмулирует нажатие сломанных 3 символьных линеек клавиш путем нажатия управляющих (1 линейка – Alt, 2 – Ctrl(L), 3 – Right Shift)+N клавиши в линейке. Клавиатура “ломается” и “восстанавливается” через 5 с.

35. Обработчик прерывания от клавиатуры, который хранит скэн-коды последних 10 нажатых клавиш и для каждой клавиши проверяет

ее наличие в этом списке. Если она там есть, то эту клавишу заменить на цифру, представляющую собой 2 младших бита скэн-кода. Список скэн-кодов может быть очищен нажатием F6.

36. Обработчик прерывания от клавиатуры, который обеспечивает следующее функционирование клавиатуры: после нажатия Ctrl-Alt-Ins все алфавитные линейки клавиатуры поочередно не работают в течение интервала t . Первоначально интервал t равен 5 секундам. Нажатие F3 может увеличивать этот интервал на 1 секунду. Возврат к нормальному функционированию повторное нажатие Ctrl-Alt-Ins.

37. С интервалом времени в 1 с блокируется цифровая клавиша. После выполнения блокировки всех цифр их можно восстановить нажатием Ctrl-Alt-Ins. Для запуска процесса блокировки цифр сначала можно использовать нажатие F1.

38. Написать обработчик прерываний от клавиатуры, которая запишет в буфер клавиатуры букву А после введения пользователем пароля. Все возможные пароли хранятся в таблице. Текущий пароль меняется каждые 10 секунд.

39. Обработчик прерывания от клавиатуры, работающий в 2 режимах, в 1 режиме – отключены цифры, во 2 – отключены буквы. Переключение между режимами каждые 5 с. Нажатие Ctrl-Alt-Ins возвращает текущий режим к стандартной обработке.

40. После нажатия Ctrl-Alt-Ins цифровая линейка клавиатуры начинает работать как пробел, 3 и 4 линейки не работают совсем.

41. Написать программу, которая функционирует в цикле с периодом 15 с следующим образом: в течение первых 5 секунд не реагирует на пробел, в течение следующих 10 с не реагирует на цифры. Нажатие Ctrl-Alt-Ins в период блокировки цифр выводит в буфер клавиатуры цифру 1, в период блокировки пробела цифру 2.

42. Написать обработчик прерываний от клавиатуры, который будет реагировать на функции алфавитно-цифровых клавиш только в течение первого периода (10 с.) цикла с периодом в 20 с. Во втором 10-секундном промежутке указанные клавиши заблокированы. Вернуться к стандартной обработке клавиш можно при нажатии Ctrl-Alt-Ins одновременно.

43. Написать обработчик прерывания от клавиатуры, который функционирует следующим образом: первые 5 с 1 линейка алфавитно-цифровой клавиатуры (qwerty...) заменяется на пробел, вторые 5 с 1 линейка отключена, третьи 5 с 1 линейка работает нормально. Увеличение интервала времени на 1 секунду – по нажатию Ctrl-Alt-Ins.

44. Обработчик прерываний от клавиатуры, работающий по следующему алгоритму: 1) при нажатии букв верхней линейки в буфер

клавиатуры заносится цифра, соответствующая порядковому номеру этой буквы в линейке; 2) Цифры заменяются на соответствующую букву верхней линейки. Этот режим начинает функционировать после Ctrl-Alt-Ins, нажатого 2 раза с интервалом не более 2 с.

45. При нажатии буквенных клавиш они удваиваются в буфере клавиатуры, цифры на основной клавиатуре заблокированы. Этот режим активен в течение каждых 30 с. из одной минуты. Оставшиеся 30 с. работает стандартный обработчик.

46. Через 1 секунду клавиша “пробел” заменяется последовательно на буквы от А до Z. После дохождения до Z, время активности буквы увеличивается на 1 с.

47. При нажатии LAlt+LCtrl осуществляется последовательная замена верхней линейки (qwerty ...) на цифры от 0 до 9 и потом на буквы А, В. После замены всей линейки замену можно отменить, нажав ESC 2 раза с интервалом не более 1 с.

48. Цифры можно набирать (от 1 до 9) только через нажатие ScrollLock, количество нажатий больше цифры на 1. Например, 3 – это 4 нажатия ScrollLock, при этом интервал между нажатиями должен быть не более 1 секунды.

49. При нажатии любой цифровой клавиши появляется последовательность 0,1,2,...9, при нажатии буквы все буквы в этой линейке заменяются на пробел. Восстановление нормального режима работы – по нажатии F1. Нормальный режим работает до нажатия F2, после чего снова действует наш режим.

50. Нажатие Ctrl-Alt-Ins позволяет выводить на экран “историю” нажатия клавиш (длина истории – 20 клавиш). Деактивировать этот режим можно двойным нажатием ScrollLock.

51. Первое нажатие Ctrl-Alt-Ins меняет местами линейку с цифрами и нижнюю линейку, второе нажатие блокирует 2 линейку, третье нажатие – обратный обмен линеек, четвертое нажатие – разблокирование 2 линейки.

52. Первые 5 секунд из 15 заблокированы буквы, вторые 5 секунд цифры заменяются на пробел, третьи 5 секунд – нормальная работа обработчика.

53. Вместо цифр реализовать занесение в буфер клавиатуры их scan-кодов. Вместо букв заносить их ASCII-коды. Этот режим активен 5 секунд из каждых 10.

54. В течение первого периода не работают гласные буквы, в течение 2 периода – цифры, в течение 3 периода – согласные буквы латинского алфавита. Величина периода первоначально составляет 5 секунд. Нажатие Enter увеличивает на 2 тика, нажатие Esc – уменьшает

период на 2 тика. Нажатие пробела отменяет работу по периодам, то есть работает стандартный обработчик.

55. При нажатии цифр выводить последовательности букв по следующему алгоритму: цифра 0 – 1 буква 'а', цифра 2 – 2 буквы 'b', цифра 3 – три буквы 'с' и т.д. Этот режим включается при нажатии клавиши F1 2 раза с интервалом между нажатиями не более 2 секунд.

Лабораторная работа № 3.

Цели и задачи работы: изучение интегрированной среды MASM32. Реализация простейших операций по управлению окнами, обработке сообщений от клавиатуры, мыши и таймера. Для работы с окнами использовать API-функции из перечня: AnimateWindow, CloseWindow, FindWindow, FlashWindow, FlashWindowEx, GetClientRect, GetParent, GetDesktopWindow, GetTitleBarInfo, GetWindowPlacement, GetWindowTextLength, IsChild, IsWindow, IsWindowVisible, MoveWindow, CloseWindow, OpenIcon, SetWindowPlacement, SetWindowText, ShowWindow, WindowFromPoint, IsChild.

Теоретические сведения о работе и пример программы приведены в [2,9].

Задание к работе: Написать приложение, осуществляющее обработку сообщений. Самостоятельно решить задачу в соответствии с индивидуальным вариантом.

Описание используемых средств для выполнения работы : язык программирования Assembler, MASM32, операционная система Windows XP/7. Используется компилятор ml, компоновщик link.

Методика выполнения работы:

1. Изучить структуру минимального приложения и логику обработки сообщений в Windows.

2. Реализовать оконное приложение, обрабатывающее сообщения в соответствии с заданием. Пример приложения с подробными комментариями приведен в [9].

3. Отладить программу.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной приведены в [9]. Защита лабораторной включает теоретические вопросы по структуре и логике работы минимального оконного приложения, предложенные преподавателем.

Примеры индивидуальных заданий:

1. После двойного щелчка левой кнопки в клиентской области окна окно начинает перемещаться по вертикали. Нажатие любой клавиши прекращает движение.

2. Щелчок правой кнопкой на клиентской области окна приводит к изменению его размеров (уменьшению в 2 раза). Нажатие любой клавиши возвращает исходные размеры.

3. Щелчок правой кнопкой на клиентской области окна приводит к изменению реакции на нажатие цифр. Каждое следующее нажатие цифры изменяет размер окна. Нажатие любой другой клавиши возвращает исходные размеры.

4. Щелчок правой кнопкой в верхней половине окна приводит к его свертке. Развертка окна – по нажатию любой функциональной клавиши.

5. Перемещение мыши в верхней половине окна приводит к появлению периодического мерцания заголовка окна. Нажатие любой буквы из линейки «QWER...» перемещает окно в правый верхний угол экрана.

6. Создать два окна, одно из которых – родительское для другого. Перемещение мыши в неклиентской области дочернего окна приводит к его свертке. Нажатие клавиши «пробел» при установке фокуса ввода на родительское окно приводит также к его свертке.

7. Щелчок левой кнопкой в неклиентской области окна перемещает окно вправо на заданное количество пикселей. Нажатие клавиши Alt вместе с цифровой клавишей возвращает окно на место.

8. Щелчок правой кнопкой в неклиентской области окна приводит к тому, что последующее нажатие любой буквенной клавиши сдвигает окно на заданное число пикселей вправо.

9. Двойной щелчок левой кнопки в рабочей области окна приводит к тому, что при нажатии цифровых клавиш окно сдвигается вниз на количество пикселей, соответствующее нажатой цифре.

10. Двойной щелчок правой кнопки в клиентской области окна приводит к тому, что при нажатии клавиш стрелок впоследствии реализуется перемещение окна по экрану.

11. Щелчок правой кнопкой в нижней половине окна разрешает изменение текста в заголовке окна. Дальнейшее нажатие букв приводит к изменению текста заголовка окна на нажатую букву.

12. Перемещение мыши в нижней половине окна максимизирует окно. Нажатие ESC – возврат окна к прежнему виду.

13. Двойной щелчок левой кнопкой в неклиентской области окна maximизирует окно. Нажатие любой буквы – возврат к прежним размерам.

14. Двойной щелчок правой кнопкой в неклиентской области окна приводит к его перемещению в цикле из его исходного состояния в горизонтальном направлении влево и вправо до тех пор, пока не будет нажата любая клавиша из нижней линейки.

15. Одновременное нажатие кнопок в неклиентской области окна приводит к его перемещению по экрану в вертикальном направлении после нажатия клавиш из линейки «ASD...».

16. Перемещение мыши в неклиентской области окна приводит к изменению его размеров. Нажатие клавиши ESC запрещает подобное изменение.

17. Двойной щелчок правой кнопки в клиентской области окна изменяет текст заголовка окна на заданный текст. Обратная замена – клавиша F1.

18. Создать два окна, одно из которых частично перекрывает другой. Щелчок правой кнопкой в левой половине верхнего окна активизирует перекрытое окно. Если до щелчка нажать клавишу «пробел», то перекрытое окно не активизируется.

19. Перемещение мыши в левой половине окна изменяет цвет фона окна (**SetClassLongPtr**). Нажатие клавиши F1 отменяет изменение.

20. Двойной щелчок левой кнопкой в неклиентской области окна свертывает его в пиктограмму. Нажатие любой функциональной клавиши разворачивает окно.

21. Создать три окна, одно из них показать. Двойной щелчок правой кнопкой в неклиентской области окна приводит к выводу на экран остальных окон, если они еще не показывались. Вывод созданных, но не показанных окон разрешен только в случае, если не нажата клавиша Enter.

22. Перемещение мыши в неклиентской области окна приводит к перемещению окна в левый верхний угол экрана. Нажатие клавиш PgUp и PgDn приводит к перемещению окна из левого верхнего угла в левый нижний угол и обратно.

23. Щелчок правой кнопкой в неклиентской области окна отменяет реакцию на нажатие клавиш End и Home, которые перемещают выведенное окно соответственно между левым верхним и правым нижним углом экрана.

24. Двойной щелчок левой кнопки в клиентской области окна приводит к смене заголовка окна. Выбор текущего заголовка окна зависит от количества нажатий клавиши Insert. Каждое следующее на-

жатие делает активным следующий по порядку заголовок из некоторого массива заголовков (циклически).

25. Вывести на экран два окна. Двойной щелчок правой кнопки в клиентской области любого из окон меняет их местами. Обратный обмен - по нажатию клавиши Delete.

26. Вывести на экран два окна. Щелчок правой кнопкой в правой половине второго окна изменяет заголовок первого. Обратный обмен – по нажатию клавиши Insert.

27. Вывести на экран три окна. Щелчок левой кнопкой в правой половине любого окна приводит к свертке в пиктограмму остальных. Обратное разворачивание, если они уже свернуты, осуществить при нажатии любой цифровой клавиши.

28. Двойной щелчок левой кнопкой в неклиентской области окна приводит к выводу на экран дочернего окна. Свернуть в пиктограмму оба окна можно при нажатии клавиш F1-F5.

29. Создать два окна, одно из них показать. Двойной щелчок правой кнопкой в неклиентской области показанного окна приводит сначала к показу второго окна, потом к его сворачиванию, потом к его разворачиванию, потом к его закрытию.

30. Создать два окна. Щелчок левой кнопкой в неклиентской области любого окна приводит к перемещению их в левый верхний угол экрана так, чтобы меньшее было сверху. Нажатие любой гласной буквы закрывает оба окна.

31. Создать два окна. Перемещение мыши в неклиентской области первого окна вызывает перемещение второго окна по горизонтали от левого к правому краю экрана и обратно. Нажатие любой цифры закрывает оба окна.

32. Двойной щелчок левой кнопки в клиентской области окна приводит к выводу на экран двух дочерних окон. По нажатию клавиши Esc заголовки дочерних окон меняются местами.

33. Вывести на экран два окна. Двойной щелчок правой кнопки в клиентской области первого окна приводит к последовательной свертке и развертке второго окна до тех пор, пока не нажата клавиша «пробел».

34. Отпускание левой кнопки мыши в верхнем левом квадранте окна изменяет текст заголовка окна на одну из нескольких строк текста в зависимости от координаты отпускания. Этот режим активен только после нажатия одной из клавиш-стрелок.

35. Щелчок левой кнопкой в верхнем левом квадранте окна приводит к его перемещению на столько пикселей влево, каков порядко-

вый номер щелчка. Максимизация окна – нажатие клавиши «стрелка вверх».

36. Двойной щелчок левой кнопкой в неклиентской области окна приводит к созданию другого окна, цвет фона которого зависит от координаты щелчка. Нажатие любой цифры закрывает созданное окно.

37. Вывести окно в правой половине экрана. Двойной щелчок правой кнопкой в неклиентской области окна приводит к тому, что каждое последующее нажатие клавиши «пробел» сдвигает окно влево.

38. Щелчок левой кнопкой в области заголовка окна изменяет этот заголовок и выводит на экран второе окно. Закрытие обоих окон происходит при нажатии любой функциональной клавиши.

39. Нажатие клавиши Insert запрещает реакцию на щелчок правой кнопкой в неклиентской области окна. Реакция на такой щелчок заключается в перемещении окна по экрану из левого нижнего в правый верхний угол.

40. Одновременный щелчок кнопок в клиентской области окна приводит к выводу на экран дочернего окна. По нажатию клавиши Home оба эти окна перемещаются к правой границе экрана.

41. Вывести на экран три окна с различными цветами фона. Двойной щелчок правой кнопки в клиентской области любого окна сворачивает в пиктограмму остальные окна. Их разворачивание – по нажатию цифр на дополнительной клавиатуре.

42. Щелчок правой кнопкой в верхнем правом квадранте окна выводит новое окно, цвет фона которого зависит от места щелчка (предусмотреть не менее 3 вариантов). Нажатие функциональной клавиши закрывает созданные окна.

43. Вывести на экран три окна, одно из которых – родительское для двух других. Щелчок левой кнопкой в верхнем правом квадранте родительского окна закрывает дочерние окна. Нажатие любой буквенной клавиши закрывает и родительское окно.

44. Вывести на экран три окна. Одновременный щелчок кнопок в неклиентской области любого окна закрывает два из них, и третье окно получает возможность перемещаться к левой границе экрана при нажатии клавиши End.

45. Двойной щелчок правой кнопкой в неклиентской области окна разрешает следующую реакцию на нажатие цифр на дополнительной клавиатуре. Каждое нажатие цифры уменьшает ширину окна на столько пикселей, какова цифра.

46. Щелчок левой кнопкой в неклиентской области окна приводит к изменению заголовка окна на новую строку в зависимости от пред-

варительного нажатия буквенных клавиш. По номеру линейки выбирается новый заголовок.

47. Щелчок правой кнопкой в неклиентской области окна вызывает его циклическое перемещение по экрану. Эта реакция на кнопку разрешается при предварительном нажатии не менее 5 буквенных клавиш.

48. Двойной щелчок левой кнопки в клиентской области окна вызывает смену заголовка окна на букву, клавиша которой была нажата перед щелчком.

49. Двойной щелчок правой кнопки в клиентской области окна выводит на экран дочернее окно, цвет фона которого зависит от координаты щелчка. Реализовать не менее 4 вариантов.

50. Щелчок правой кнопкой в нижнем левом квадранте окна сворачивает его в пиктограмму, а нажатие клавиши F8 разворачивает окно при условии, что перед этим была нажата хотя бы одна цифра.

51. Щелчок левой кнопкой в нижнем левом квадранте окна разрешает перемещение окна по экрану с помощью клавиш-стрелок, PgUp, PgDn.

52. Двойной щелчок левой кнопкой в неклиентской области окна позволяет в дальнейшем при нажатии клавиш 1-4 двигать окно по экрану в четырех направлениях.

53. Двойной щелчок правой кнопкой в неклиентской области окна выводит на экран два дочерних окна с разными стилями. Нажатие любой функциональной клавиши закрывает одно из дочерних окон.

54. Щелчок левой кнопкой в неклиентской области окна выводит длину заголовка окна. Нажатие любой буквы меняет заголовок на эту букву.

55. Щелчок правой кнопкой в неклиентской области окна разрешает перемещение окна в левый верхний угол (по нажатии F1), левый нижний угол (F2), правый верхний (F3), правый нижний (F4).

56. Двойной щелчок левой кнопки в клиентской области окна разрешает создание дочернего окна по нажатию цифры. Заголовком дочернего окна должна стать эта цифра.

57. Двойной щелчок правой кнопки в клиентской области окна закрывает все дочерние окна, созданные по нажатии клавиш F1-F4.

58. Щелчок правой кнопкой в нижнем правом квадранте окна разрешает перемещение окна при нажатии стрелок на дополнительной клавиатуре.

59. Щелчок левой кнопкой в нижнем правом квадранте окна приводит к созданию дочернего окна, заголовок которого – буква, нажатая перед щелчком.

60. Двойной щелчок левой кнопкой в неклиентской области окна выводит на экран другое окно. Щелчок правой кнопки в его клиентской области разрешает его перемещение вверх-вниз по нажатию F4.

61. Двойной щелчок правой кнопкой в неклиентской области окна разрешает перемещение окна по экрану с помощью клавиш: Т – вверх, В – вниз, L – влево, R – вправо.

62. Щелчок левой кнопкой в неклиентской области окна приводит к созданию двух окон с разными стилями класса. Нажатие клавиш-стрелок позволяет перемещать любое окно, если при активном 1 окне нажата любая гласная буква.

63. Вывести на экран 2 окна. Щелчок левой кнопкой в неклиентской области любого окна изменяет заголовок второго окна. Нажатие любой цифры возвращает исходный заголовок.

64. Щелчок правой кнопкой в неклиентской области окна разрешает вывод двух дочерних окон. Первое окно выводится на экран при нажатии буквы «А», второе – при нажатии буквы «В». Закрытие всех окон – по нажатию «ESC».

65. Двойной щелчок левой кнопки в клиентской области окна выводит дочернее окно. Закрытие дочернего окна – нажатие F1, повторный вывод – нажатие F2.

66. Двойной щелчок правой кнопки в клиентской области окна разрешает перемещение окна влево, вправо, вверх и вниз соответственно при нажатии F1, F2, F8, F9.

67. Двойной щелчок левой кнопки в неклиентской области окна выводит на экран два дочерних окна. Нажатие любой цифры закрывает дочерние окна.

68. Двойной щелчок левой кнопки в неклиентской области окна приводит к появлению на экране дочернего окна. Заголовок этого окна может изменяться на букву – в зависимости от места щелчка.

69. Щелчок правой кнопки в клиентской области окна разрешает перемещать окно по экрану при нажатии клавиш-стрелок. Двойной щелчок левой кнопки запрещает такое перемещение.

70. Вывести на экран родительское окно и два дочерних. Щелчок левой кнопкой в клиентской области любого дочернего окна разрешает их перемещение по экрану вверх и вниз при нажатии букв H и D соответственно.

71. Двойной щелчок правой кнопкой в неклиентской области окна разрешает создание дочернего окна при нажатии цифр 0 или 1. Нажатие любой буквы уменьшает размеры дочернего окна в 2 раза.

72. В левом верхнем углу клиентской области окна создать дочернее окно. После нажатия левой клавиши мыши 4 раза «мигает» временное окно, а после нажатия правой – 3 раза «мигает» дочернее окно.

73. Создать окно приложения размером в одну шестнадцатую площади экрана с заголовком «Идет форматирование диска» без кнопок изменения размеров, закрытия и сворачивания в пиктограмму и без кнопки системного меню. При перемещении курсора мыши над клиентской областью окно должно «убегать» от курсора мыши в случайном образом выбранном направлении, оставаясь в пределах экрана.

74. В левом верхнем углу клиентской области окна создать временное окно площадью в одну шестнадцатую площади этой области. При нажатии на левую (правую) клавишу мыши временное окно переместить в соседний по ходу (против хода) часовой стрелки угол клиентской области.

75. При запуске 2-го экземпляра приложения спросить пользователя, нужно ли его запустить. Если пользователь ответит «Да», то запустить его. Иначе изменить заголовок 1-го экземпляра.

76. Окно приложения без заголовка занимает весь экран фоном рабочего стола. Закрытие окна по щелчку правой кнопки мыши.

77. В центре клиентской области окна располагается невидимое окно без заголовка размером в четверть площади клиентской области. После нажатия левой клавиши мыши над клиентской областью любого из окон окно без заголовка должно стать видимым, а после нажатия правой – невидимым.

78. Создать окно размером в четверть площади экрана. После двойного щелчка мыши окно перемещается так, что его центр совпадает с координатами курсора мыши в момент щелчка.

79. Дочернее окно размером в $\frac{1}{4}$ родительского окна при перемещении курсора мыши над ним «убегает» от курсора мыши в произвольном направлении, оставаясь в пределах клиентской области родительского окна.

80. При запуске не первого экземпляра приложения выдать предупреждающее сообщение о количестве уже работающих копий этого приложения. Запустить экземпляр, только если согласен пользователь.

Лабораторная работа № 4.

Цели и задачи работы: изучение и практическое использование функций GDI для работы с устройствами вывода информации.

Теоретические сведения о работе и пример программы приведены в учебном пособии [2,9].

Задание к работе: Написать приложение, осуществляющее:

1. Вывод текста с использованием не менее чем двух типов шрифтов.
2. Вывод графических примитивов (прямоугольники, эллипсы, дуги и т.п.) с использованием различных перьев и кистей для заливки;
3. Вывод мини-мультфильма по сюжету. Сюжет мультипликации придумывается студентом и согласовывается с преподавателем с целью одинакового уровня сложности различных вариантов сюжетов. Самостоятельно решить задачу в соответствии с индивидуальным вариантом.
4. Изучить средства загрузки DLL, оформить 2-3 функции вывода данных на графическое устройство как DLL. Функции для библиотeki подобрать самостоятельно.

Описание используемых средств для выполнения работы : язык программирования Assembler, MASM32, операционная система Windows XP/7. Используется компилятор ml, компоновщик link.

Методика выполнения работы:

1. Изучить функции GDI Windows.
2. Подготовить спрайты для мини-мультфильма.
3. Реализовать оконное приложение согласно заданию.
4. Отладить программу.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной приведены в [9]. Защита лабораторной включает теоретические вопросы по функциям и понятиям GDI, предложенные преподавателем.

Примеры индивидуальных заданий для мультипликации:

1. Море с видимой линией горизонта и плывущий кораблик.
2. Распускающийся цветок.
3. Кот, бегущий за мышкой.
4. Домик, из трубы которого идет дым.
5. Снеговик, тающий после появления солнышка.
6. С яблони падает яблоко.
7. Солнечное затмение.
8. На ясном небе появляется тучка, идет дождь, затем появляется радуга.
9. Золотая рыбка плавает в аквариуме.
10. В прозрачный стакан капает вода из крана, уровень воды увеличивается.
11. Полет ракеты в космосе.

12. Движущийся автомобиль с мигалкой.
13. Карандаш, рисующий несколько разноцветных линий.
14. Летящая бабочка.
15. Цыпленок, клюющий зерно.
16. Полет Винни-Пуха на воздушном шарике.
17. Прыжок человека с парашютом с самолета.
18. Полет летающей тарелки с изменением ее цвета.
19. Человек, перепрыгивающий через препятствие.
20. Часы с движущимися стрелками.
21. Часы с электронным циферблатом.
22. Часы с кукушкой.
23. Паровоз тянет вагон, из трубы идет дым.
24. Всплывающая подводная лодка.
25. Закат солнца.

Лабораторная работа № 5.

Цели и задачи работы: изучение и практическое использование API-функций для обработки информации файловой системы одного из разделов жесткого диска. Использование ресурса меню, элементов управления форм.

Теоретические сведения о работе приведены в учебных пособиях [9,10], в источниках [3,4,6,7].

Задание к работе: реализовать приложение, позволяющее получить информацию о файловой системе компьютера, и выполнить указанные операции с файлами и каталогами. Оформить выбор действий пользователя с применением ресурса меню. Ввод-вывод информации реализовать с помощью элементов управления форм. Используемые API-функции: CreateFile, CloseHandle, CreateDirectory(Ex), CopyFile(Ex), DeleteFile, FindClose, FindFirstFile(Ex), FindNextFile, GetFileAttributes(Ex), GetFileInformationByHandle, GetFileSize, GetFileTime, GetFileType, MoveFile(Ex), RemoveDirectory, SetFileAttributes, SetFileTime, FileTimeToSystemTime, GetComputerNameEx, GetCurrentDirectory, GetDiskFreeSpaceEx, GetLogicalDrives, GetSystemDirectory, GetVolumeInformation, SetCurrentDirectory.

Описание используемых средств для выполнения работы : язык программирования Assembler, MASM32, операционная система Windows XP/7, Linux, язык программирования C.

Методика выполнения работы:

1. Изучить теоретический материал по обработке данных файловой системы.

2. Реализовать приложение согласно заданию в различных ОС.
3. Отладить программы.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями, тесты.

Контрольные вопросы по лабораторной работе включают вопросы, приведенные в [10], а также охватывают лекционный материал по теме. Защита лабораторной включает теоретические вопросы, предложенные преподавателем.

Примеры индивидуальных заданий :

1. Вывести список системных файлов для выбранного каталога. Первый найденный системный файл скопировать в новый каталог.

2. Вывести список файлов с атрибутом «только чтение» для выбранного каталога. Последний найденный файл с данным атрибутом скопировать в новый каталог.

3. Сформировать список подкаталогов для выбранного каталога. Первый найденный подкаталог с содержимым скопировать в новый каталог.

4. Сформировать список файлов и подкаталогов для выбранного каталога. Первый найденный подкаталог удалить вместе со всем содержимым, предварительно запросив у пользователя подтверждение на удаление.

5. Для выбранного каталога сформировать список содержащихся в нем файлов, дата создания которых принадлежит заданному временному промежутку. Переместить каталог в пределах тома, если список оказался пуст.

6. Для заданного каталога сформировать список файлов, длина которых не превышает заданное число. Первый найденный файл подходящей длины удалить, запросив подтверждение.

7. Определить для файла, заданного полным путем, дату его создания. Если файл создан раньше указанной даты, то скопировать его, а иначе переместить в каталог-родитель.

8. Определить для файла, заданного полным путем, время его создания и последнего обновления. Если файл создан позже указанной даты, то удалить его, если он не является системным или временным, иначе переместить в новый каталог.

9. Определить для файла, заданного именем, каталог, который является для него родительским. Если родительский каталог создан раньше указанной даты, то скопировать его содержимое в новый каталог.

10. Найти на логическом диске файлы, созданные не более двух суток назад и вывести их имена. Если среди них есть файлы, начинающиеся на букву «а», скопировать их в заданный каталог.

11. Найти на логическом диске каталоги, содержащие не более трех файлов и подкаталогов, и вывести их имена. Первый найденный такой каталог скопировать со всем содержимым в новый подкаталог корневого каталога.

12. Задано имя файла. Определить имена каталогов, в которых он содержится, и, если найден не один каталог-родитель, то все файлы с указанным именем поместить в новый каталог с известным именем.

13. Определить количество доступных логических дисков. С одного из них вывести все файлы с атрибутом «только чтение». Файлы, длина которых превышает заданное число, скопировать в новый каталог, изменив данный атрибут.

14. Задано 2 имени файлов. Определить, какой из них создан раньше, и тот, который создан позже, скопировать в новый каталог.

15. Задано 2 имени файлов. Определить, родительский каталог которого из них создан позже, и тот каталог, который создан позже, скопировать в новый каталог и сделать его текущим.

16. Найти все каталоги, не содержащие файлов с расширением txt, sys и asm. Вывести на экран атрибуты последнего из них, а затем изменить его дату создания на заданную дату.

17. Сосчитать количество подкаталогов в текущем каталоге. Если таких каталогов больше заданного числа, вывести атрибуты самого длинного файла. Иначе скопировать содержимое текущего каталога в новый и сделать его текущим.

18. Выдать имена всех файлов, содержащихся в каталогах, созданных после указанной даты. Последний найденный файл переместить в заданный каталог.

19. На заданном логическом диске найти имена всех файлов, если в каталогах-предках есть системные файлы. Первый найденный файл скопировать в новый каталог с заданным именем.

20. На заданном логическом диске найти имена всех файлов, соседями по каталогу для которых являются файлы с атрибутом «только чтение». Каждый второй найденный файл скопировать во временный каталог, который после просмотра его содержимого уничтожить.

21. Найти файлы, созданные до указанной даты, и вывести его атрибуты на экран при условии, что у него есть не менее двух подкаталогов-предков в полном пути. Последний найденный файл скопировать в заданный каталог.

22. Для данного диска выдать имена всех каталогов, содержащие файлы с расширением "exe" или "com". Если в каталоге более трех файлов с таким расширением, переместить во временный новый каталог очередной файл из заданного каталога. Очередность файлов сдвигается после каждого перемещения.

23. Найти файлы, созданные после указанной даты, и вывести их атрибуты на экран при условии, что у них есть не менее трех подкаталогов-предков в полном пути. Первый найденный файл скопировать в заданный каталог.

24. Найти имена всех каталогов, не содержащих файлов с расширением "sys" или "c" и созданных после указанной даты. Содержимое таких каталогов переместить во временный новый каталог, который удалить после просмотра.

25. Найти имена всех каталогов, не содержащих файлов с расширением "exe" или "com" и созданных до указанной даты. Содержимое таких каталогов переместить во временный новый каталог, который удалить после просмотра. Содержимое этого каталога переместить обратно.

26. Определить имя компьютера, где запущено приложение. Определить имя системного каталога ОС, сделать его текущим, а затем вывести на экран имена всех файлов, длина имени которых не превышает заданного значения. Последний найденный файл скопировать в новый каталог.

27. Найти файлы, у которых каталог-родитель содержит пустой каталог. Если таких файлов несколько, переместить их после подтверждения в новый каталог.

28. Определить свободное дисковое пространство на томе. Затем найти все файлы, которые занимают больше, чем заданное количество секторов. Найденные файлы скопировать после подтверждения в каталог с заданным именем.

29. Определить имя компьютера, где запущено приложение. Определить имя системного каталога ОС, затем вывести на экран имена всех его файлов, длина которых не превышает заданного значения. Последний найденный файл переместить в новый каталог по согласованию с пользователем.

30. Выдать имена всех каталогов заданного логического диска, содержащих более одного файла с расширением "doc". Последний найденный каталог сделать текущим и скопировать в него три файла из системного каталога ОС.

31. Выдать имена всех файлов с заданным расширением, с заданного логического диска с указанием полного пути. Если размер хотя

бы одного из них совпадет с заданным числом, то переместить этот файл в новый каталог.

32. Найти каталоги, содержащие более 2-х файлов с датой создания не позже, чем дата создания заданного файла. Если таких каталогов не один, то переместить содержимое первого из них в новый временный каталог, который удалить после просмотра его содержимого.

33. Выдать имена всех каталогов уровня вложенности более 3, начиная от корневого, содержащего файлы с заданным расширением. Последний найденный каталог удалить после подтверждения пользователя.

34. Для заданного логического диска выдать имена всех пустых каталогов. В последний найденный каталог скопировать первые 5 файлов из системного каталога ОС.

35. Сосчитать количество подкаталогов в текущем каталоге. Если таких каталогов меньше заданного числа, вывести атрибуты самого короткого файла из текущего каталога. Иначе скопировать содержимое текущего каталога в новый и сделать его текущим

36. Для указанного каталога выдать имена всех содержащихся в нем (именно в нем, а не в его подкаталогах) файлов. Все эти файлы переместить после подтверждения пользователя в заданный каталог.

37. Выдать имена файлов, имя которых содержит не более 3 символов (без расширения) и в родительском каталоге которых есть еще хотя бы один каталог. Для последнего из них установить дату создания 01.01.01.

38. Напечатать имена всех файлов, начинающихся с буквы "а", находящихся не в корневой директории и являющихся не зашифрованными. Для последнего найденного файла установить заданную дату создания по согласованию с пользователем.

39. Выдать имена каталогов, не содержащих файлов (каталоги могут быть). Если такие каталоги есть, то переместить содержимое одного из них в другой временный каталог.

40. На заданном ЛД выдать имена всех каталогов, чей уровень вложенности превышает заданный. Если такие каталоги отсутствуют, скопировать все файлы из корневого каталога (не содержимое каталогов!) в новый каталог.

41. Заданы имя каталога и имя файла. Если файл занимает более заданного количества секторов, то выдать содержимое каталога, иначе выдать список атрибутов файла. Файл уничтожить по согласованию с пользователем.

42. Вывести на экран содержимое тех каталогов, у которых длина имени не превышает заданное количество букв и которые не содержат

файлов с расширением *.txt. Если таких каталогов больше 2, то переместить их файлы по согласованию с пользователем в другой каталог.

43. Выдать имена всех файлов, созданных ранее указанной даты, а затем переместить их во временный каталог. После просмотра временного каталога реализовать обратное перемещение.

44. Выдать содержимое каталогов, созданных до указанной даты и содержащих не менее заданного количества файлов и подкаталогов. Если таких каталогов нет, то получить информацию о томе и файловой системе.

45. Определить для файла, заданного полным путем, время его создания и последнего обновления. Если файл создан раньше указанной даты, то удалить его по согласованию с пользователем, если он не является системным, иначе переместить в новый каталог.

46. Определить для файла, заданного полным путем, каталог, который является для него родительским. Если родительский каталог создан позже указанной даты, то скопировать его содержимое в новый каталог.

47. Найти файлы, созданные не более недели назад и вывести их имена. Если среди них есть файлы, начинающиеся на букву «i», скопировать их в заданный каталог.

48. Найти на логическом диске каталоги, содержащие не более трех файлов и подкаталогов, и вывести их имена. Последний найденный такой каталог скопировать со всем содержимым в новый подкаталог корневого каталога.

49. Задано имя файла. Определить имена каталогов, в которых он содержится, и, если найден только один каталог-родитель, то все его файлы переместить в новый каталог с известным именем по согласованию с пользователем.

50. Определить количество доступных логических дисков. С одного из них вывести все файлы с атрибутом «архивный» или «только чтение». Файлы, длина которых не превышает заданное число, скопировать в новый каталог, изменив данный атрибут.

51. Сосчитать общее количество подкаталогов в заданном, и содержимое последнего из них переместить в новый каталог.

52. Сосчитать количество файлов, дата создания которых находится в заданном диапазоне. Первый из таких файлов удалить, запросив у пользователя подтверждение.

53. Найти каталог, который содержит хотя бы 1 пустой каталог и не менее 2 файлов с заданной датой создания. Если такого нет, то из текущего каталога удалить файл по согласованию с пользователем.

54. Сосчитать количество каталогов, которые созданы до указанной даты и содержат не менее одного пустого каталога. Если таких каталогов нет, по согласованию с пользователем удалить первый найденный файл, начинающийся на букву «а».

55. Сосчитать количество папок, которые содержат не менее трех файлов с расширением «exe». Если таких папок нет, то переместить файл с заданным именем в каталог на один уровень выше, чем родительский для заданного файла. Если такого каталога нет, что сообщить.

56. Найти файлы, у которых каталог-родитель содержит пустой каталог. Последний найденный файл удалить по согласованию с пользователем.

57. Найти файлы, все каталоги-родители которых начинаются на заданную букву. Последний найденный файл переместить в заданный каталог.

58. Найти все файлы, для которых дата создания находится в заданном диапазоне и все каталоги-предки содержат в имени не более 2 символов. Последний найденный файл удалить по согласованию с пользователем.

59. Для заданного каталога вывести имена всех подкаталогов, содержащих не менее заданного количества файлов. Если таких нет, то скопировать содержимое последнего просматриваемого каталога в новую папку.

60. Для заданного каталога и заданного файла определить все родительские каталоги для каждого. Если в двух полученных списках есть каталоги с одинаковыми именами, то удалить файл по согласованию с пользователем.

61. Для заданного файла определить его длину и дату создания. Если длина меньше заданного числа, то скопировать файл в заданный каталог. Иначе установить дату создания 07.07.07.

62. Для файла, заданного полным путем, определить, имеются ли в заданном каталоге другие файлы с тем же именем, но меньшей длиной. Если такие файлы есть, то скопировать их в новый каталог.

63. Для файла, заданного полным путем, определить дату создания. Если файл создан позже указанной даты, скопировать его в заданный каталог.

64. Для файла, заданного полным путем, проверить, имеются ли в его родительском каталоге файлы с расширением exe. Если есть, то скопировать их в новый каталог.

65. Сосчитать количество подкаталогов в заданном каталоге. Если их больше заданного числа, то скопировать самый длинный файл из заданного каталога в первый найденный каталог на диске С.

Лабораторная работа № 6.

Цели и задачи работы: изучение и практическое использование синхронного и асинхронного ввода-вывода данных при обмене с периферийными устройствами.

Теоретические сведения о работе приведены в учебном пособии [14] и конспекте лекций.

Задание к работе:

1. Изучить теоретический материал по синхронному и асинхронному вводу-выводу.
2. Написать приложение, осуществляющее асинхронное чтение/запись данных с использованием функций WaitForSingleObject, GetOverlappedResult, CreateFile, ReadFile, WriteFile.
3. Изучить работу с высокоуровневыми компонентами обмена с устройствами через интерфейсы COM и USB.

Описание используемых средств для выполнения работы : язык программирования Assembler, MASM32, Visual Studio, операционная система Windows XP/7.

Методика выполнения работы:

1. Изучить функции для асинхронного ввода-вывода.
2. Реализовать приложение согласно заданию 2.
3. Отладить программу.
4. Изучить реализацию обмена с устройствами через USB и COM интерфейсы, настройку свойств компонентов для программирования обмена.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст программы с комментариями.

Контрольные вопросы по лабораторной приведены в [10]. Защита лабораторной включает теоретические вопросы, предложенные преподавателем, и демонстрацию настройки компонентов для обмена по USB и COM интерфейсам.

Лабораторная работа № 7.

Цели и задачи работы: изучение структуры драйверов устройств, обрабатываемых событий.

Теоретические сведения о работе приведены в источниках [5,8,11] и в конспекте лекций по дисциплине.

Задание к работе:

1. Изучить структуру драйвера для клавиатуры.
2. Разработать модификацию функций драйвера для выполнения фильтрации на основе примера из WDK.

Описание используемых средств для выполнения работы : Visual Studio 2010, WDK, операционная система Windows XP/7.

Методика выполнения работы:

1. Изучить теоретический материал по архитектуре драйверов
2. Изучить примеры и документацию из WDK.
3. Предложить модификацию функций драйвера.

Требования к отчету:

Отчет должен содержать титульный лист, задание, текст модифицируемых функций драйвера с комментариями.

Контрольные вопросы по лабораторной работе включают лекционный материал по теме. Защита лабораторной работы включает теоретические вопросы, предложенные преподавателем.

Список литературы:

1. Аблязов, Р.З. Программирование на ассемблере на платформе x86-64. – СПб.: ДМК-ПРЕСС, 2011. – 304 с.
2. Ганеев, Р.М. Проектирование интерфейса пользователя средствами Win32 API: [учеб. пособие для вузов].-М.: Горячая линия -Телеком, 2007.-357 с.: ил. Режим доступа http://elib.altstu.ru/elib/liter/Ganev_win32_api.zip.
3. Таненбаум Э. Архитектура компьютера. – ПИТЕР, 2010. – 844 с.
4. Несвижский В. Программирование аппаратных средств в Windows. - СПб.: БХВ-Петербург, 2008. – 528 с.
5. Пенни Орвик, Гай Смит. Windows Driver Foundation. Разработка драйверов. – Изд-во БХВ-Петербург, 2008. – 880 с.
6. Пирогов В. Ассемблер для Windows.- СПб.: БХВ-Петербург, 2007. – 896 с.
7. Роберт Лав. Linux. Системное программирование. – СПб.:«Питер», 2008. – 416 с.
8. Солдатов В.П. Программирование драйверов под Windows. – Изд-во БИНОМ-ПРЕСС, 2006. – 576 с.
9. Сучкова, Л.И. Win32 API: основы программирования: учебное пособие/ Л.И.Сучкова; Алт. гос. техн. ун-т им. И.И. Ползунова. – Барнаул: АлтГТУ, 2010. – 138 с. Режим доступа http://elib.altstu.ru/elib/eum/vsib/Suchkova_WinAPI.pdf
- 10.Сучкова, Л.И. Программно-аппаратные аспекты низкоуровневого обмена с периферийными устройствами: учеб.пособие/ Л.И. Сучкова; Алт. гос. техн. ун-т им. И.И. Ползунова, – Барнаул, 2012. – Режим доступа <http://elib.altstu.ru/>
- 11.Уолтер Они. Использование Microsoft Windows Driver Model. – СПб.: ПИТЕР, 2007. – 764 с. -1 экз.

Лариса Иннокентьевна Сучкова

**Учебно-методическое пособие по дисциплине
«Периферийные устройства»**

Издано в авторской редакции

Подписано в печать 25.05.2012. Формат 60х84 1/16.

Печать – цифровая. Усл.п.л. 2,09 .

Тираж 25 экз. Заказ 2012 - 326

Отпечатано в типографии АлтГТУ,
656038, г. Барнаул, пр-т Ленина, 46
тел.: (8–3852) 36–84–61

Лицензия на полиграфическую деятельность
ПЛД №28–35 от 15.07.97 г.