**Лабораторная работа 4(3). Консоль управления моделью Simple Computer. Псевдографика. «Большие символы».**

**Работа выполнена студентами группы ИП-813: Бурдуковский И.А; Стояк Ю.К.**

Цель работы: Изучить работу текстового терминала с псевдографическими символами. Понять, что такое шрифт и как он используется в терминалах при выводе информации. Разработать библиотеку myBigChars, реализующую функции по работе с псевдографикой и выводу «больших символов» на экран. Доработать консоль управления Simple Computer так, чтобы выводились псевдографические элементы.

Были разработаны следующие функции:

int bc\_printA (char \* str) - выводит строку символов с использованием дополнительной кодировочной таблицы;

int bc\_printA(char \*str){

printf("\033(0%s\033(B",str);

return 0;

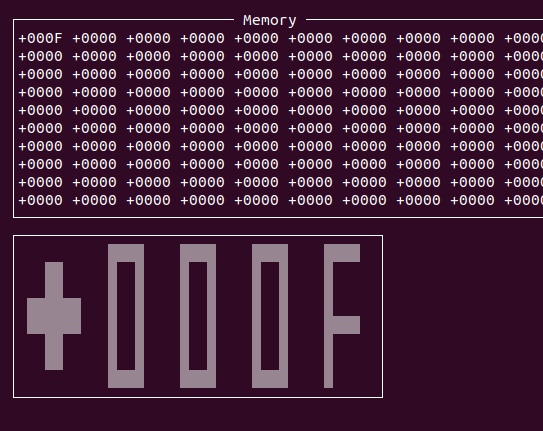
}

int bc\_box(int x1, int y1, int x2, int y2) - выводит на экран псевдографическую рамку, в которой левый верхний угол располагается в строке x1 и столбце y1, а еѐ ширина и высота равна y2 столбцов и x2 строк;

int bc\_box(int x1, int y1, int x2, int y2){

|  |
| --- |
|  |
|  | if(x1<0 || y1<0 || x2<0 || y2<0){ |
|  | cout << "Invalid cols and rows"; |
|  | return 2; |
|  | } |
|  | mt\_gotoXY(x1,y1); |
|  | bc\_printA("l"); |
|  | for(int i=y1;i<y1+y2;i++){ |
|  | bc\_printA("q"); |
|  | } |
|  | bc\_printA("k"); |
|  | for(int i=x1;i<x1+x2;i++){ |
|  | mt\_gotoXY(i+1,y1); |
|  | bc\_printA("x"); |
|  | mt\_gotoXY(i+1,y1+y2+1); |
|  | bc\_printA("x"); |
|  |  |
|  | } |
|  | mt\_gotoXY(x1+x2+1,y1); |
|  | bc\_printA("m"); |
|  | for(int i=y1;i<y1+y2;i++){ |
|  | bc\_printA("q"); |
|  | } |
|  | bc\_printA("j"); |
|  | return 0; |
|  | } |

int bc\_printbigchar (int [2], int x, int y, enum color, enum color) - выводит на экран "большой символ" размером восемь строк на восемь столбцов, левый верхний угол которого располагается в строке x и столбце y. Третий и четвѐртый параметры определяют цвет и фон выводимых символов. "Символ" выводится исходя из значений массива целых чисел следующим образом. В первой строке выводится 8 младших бит первого числа, во второй следующие 8, в третьей и 4 следующие. В 5 строке выводятся 8 младших бит второго числа и т.д. При этом если значение бита = 0, то выводится символ "пробел", иначе - символ, закрашивающий знакоместо (ACS\_CKBOARD);



Справа приложен результат работы функций.

int bc\_printbigchar(int \*big, int x, int y, enum Colors fg, enum Colors bg){

|  |
| --- |
|  |
|  | if ((x < 0) || (y < 0)) |
|  | return -1; |
|  | mt\_gotoXY(x, y); |
|  | mt\_setfgcolor(fg); |
|  | mt\_setbgcolor(bg); |
|  | int position = 0; |
|  | char row[8]; |
|  | row[8] = '\0'; |
|  | for (int i=0; i<8; i++){ |
|  | for(int j=0; j<8; j++){ |
|  | int bit = (big[position] >> ((i % 4) \* 8 + j)) & 1; |
|  | if(bit == 0) |
|  | row[j] = ' '; |
|  | else |
|  | row[j] = 'a'; |
|  | } |
|  | mt\_gotoXY(x+i, y); |
|  | bc\_printA(row); |
|  | if(i==3){ |
|  | position = 1; |
|  | } |
|  | } |
|  | mt\_setbgcolor(DEFAULT); |
|  | mt\_setfgcolor(DEFAULT); |
|  | return 0; |
|  | } |

int bc\_setbigcharpos (int \* big, int x, int y, int value) - устанавливает значение знакоместа "большого символа" в строке x и столбце y в значение value;

int bc\_getbigcharpos(int \* big, int x, int y, int \*value) - возвращает значение позиции в "большом символе" в строке x и столбце y;

int bc\_setbigcharpos (int \*big, int x, int y, int value){

|  |
| --- |
|  |
|  | if (x < 0 || x > 7 || y < 0 || y > 7 || value < 0 || value > 1) |
|  | return -1; |
|  | int position; |
|  | if (y > 3) |
|  | position = 1; |
|  | else |
|  | position = 0; |
|  | y = y % 4; |
|  | if (value == 0) |
|  | big[position] &= ~(1 << (y\*8 + x)); |
|  | else |
|  | big[position] |= 1 << (y\*8 + x); |
|  |  |
|  | return 0; |
|  | } |
|  |  |
|  | int bc\_getbigcharpos(int \*big, int x, int y, int \*value){ |
|  | int position; |
|  | if (x < 0 || x > 7 || y < 0 || y > 7) |
|  | return -1; |
|  | if (y > 3) |
|  | position = 1; |
|  | else |
|  | position = 0; |
|  | y = y % 4; |
|  | \*value = (big[position] >> (y\*8 + x)) & 1; |
|  |  |
|  | return 0; |
|  | } |

int bc\_bigcharwrite (int fd, int \* big, int count) - записывает заданное число "больших символов" в файл. Формат записи определяется пользователем;

int bc\_bigcharread (int fd, int \* big, int need\_count, int \* count) считывает из файла заданное количество "больших символов". Третий параметр указывает адрес переменной, в которую помещается количество считанных символов или 0, в случае ошибки.

int bc\_bigcharwrite(int fd, int \*big, int count){

|  |
| --- |
|  |
|  | if (write(fd, big, count \* (sizeof(int)) \* 2) == -1) |
|  | return -1; |
|  | return 0; |
|  | } |
|  |  |
|  | int bc\_bigcharread(int fd, int \*big, int need\_count, int \*count){ |
|  | \*count = read(fd, big, need\_count \* sizeof(int) \* 2); |
|  | if (\*count == -1){ |
|  | \*count = 0; |
|  | return -1; |
|  | } |
|  | \*count = \*count / (sizeof(int) \* 2); |
|  | return 0; |
|  | } |

int bc\_setbigcharpos и int bc\_getbigcharpos не использовались в нашей программе, так как были обнаружены 16 ричные значения для более удобной отрисовки половин big char’a. Ниже приложены их кодировки:

static int bc\_symbols[17][2] = {{0x2424243C, 0x3C242424}, // 0

|  |
| --- |
|  |
|  | {0x18181818, 0x18181818}, // 1 |
|  | {0x2020203C, 0x3C04043C}, // 2 |
|  | {0x2020203C, 0x03C20203C}, // 3 |
|  | {0x3C242424, 0x20202020}, // 4 |
|  | {0x404043C, 0x03C20203C}, // 5 |
|  | {0x404043C, 0x3C24243C}, // 6 |
|  | {0x2020203C, 0x20202020}, // 7 |
|  | {0x3C24243C, 0x3C242424}, // 8 |
|  | {0x3C24243C, 0x3C202020}, // 9 |
|  | {0x24242418, 0x2424243C}, // A |
|  | {0x1C24241C, 0x1C24241C}, // B |
|  | {0x404043C, 0x3C040404}, // C |
|  | {0x2424241C, 0x1C242424}, // D |
|  | {0x3C04043C, 0x3C040404}, // E |
|  | {0x404043C, 0x404043C}, // F |
|  | {0x7E181800, 0x18187E}}; // + |

**Результат работы программы:**

