

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»

Лабораторная работа №1
«Часть 3 – Разработка программ»

Выполнил: студент 4 курса

ИВТ, гр. ИП-713

Михеев Н.А.

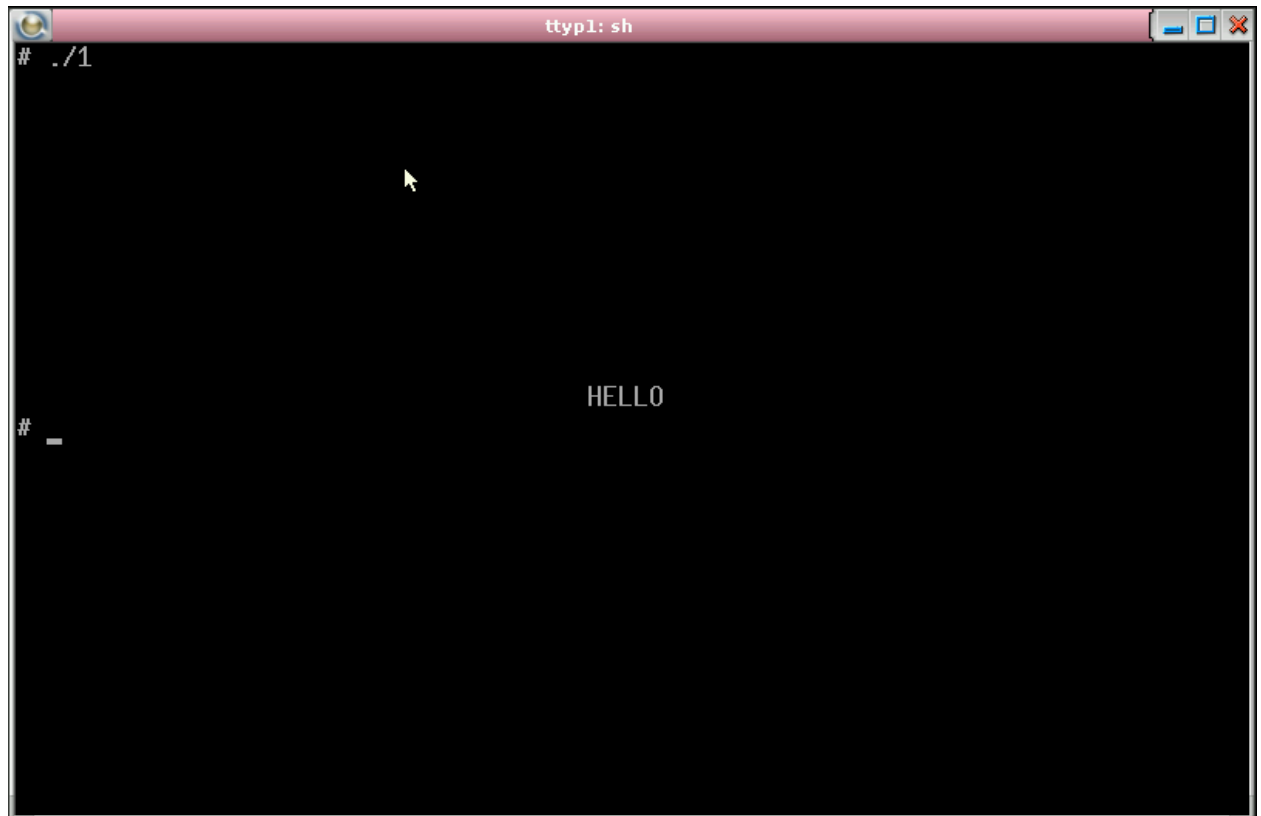
Проверил: старший преподаватель кафедры ПМиК

Милешко А.В.

Новосибирск, 2020 г.

1. Написать программу, выводящую сообщение "HELLO" в центре чистого экрана.

Для выполнения задания была написана программа с использованием esc-последовательности: «\033[u;xH» - установка курсора в позицию (x,y). Так как стандартный размер терминала 80 на 25, то отправим курсор на позицию пополам от этих значений: по x – 13, по y – 37 (не ровно 40, чтобы слово стояло ровнее, три буквы слева от середины и две буквы справа). Результат работы программы:



Код программы будет выложен ниже в приложении под заголовком “1.с”.

2. Написать программу, позволяющую определять коды нажимаемых клавиш и восстанавливающую исходный вид терминала (цвет, курсор) при выходе.

Для выполнения задания была разработана программа, меняющая цвет фона терминала на зеленый, а цвет символов на синий, а терминал переключается в raw-режим. Далее в программе стартует бесконечный цикл до тех пор, пока не будет нажат enter. В цикле идет ожидание ввода символа, когда символ нажат – выводится нажатый символ и его код. Если необходимо закрыть программу – нажимается Enter, происходит остановка цикла и возврат терминала к начальному состоянию.

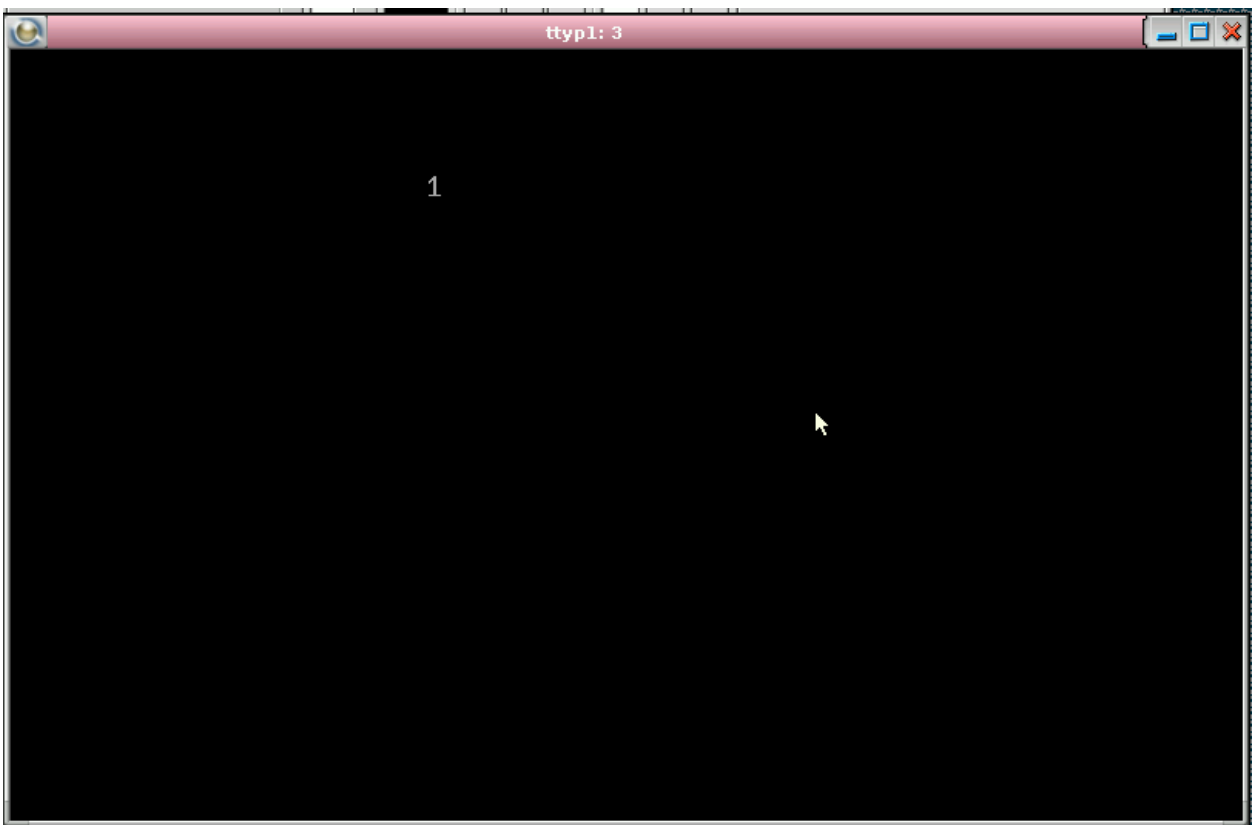
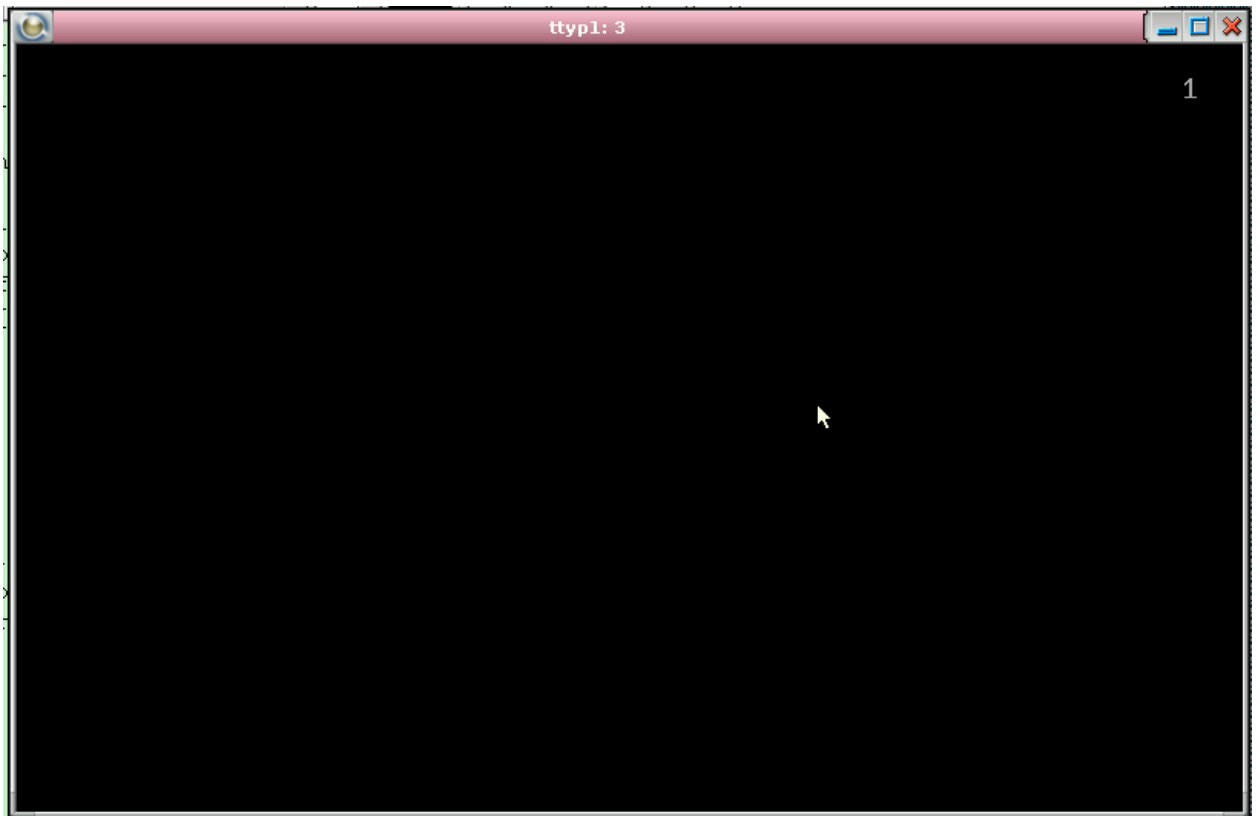
```
ttyp1: sh
# cc 2.c -o 2
# ./2
Written symbol=a, code=97
Written symbol=s, code=115
Written symbol=A, code=65
Written symbol=B, code=66
Written symbol=3, code=51
Written symbol=5, code=53
Written symbol=$, code=36
Written symbol=@, code=64
Written symbol=Z, code=37
Written symbol=^, code=94
Written symbol={, code=123
Written symbol=}, code=125
Written symbol=[, code=91
Written symbol=], code=93
Written symbol=ode=27
Written symbol=[, code=91
Written symbol=@, code=64
Written symbol=
      , code=10

# echo test
test
#
```

Код программы будет выложен ниже в приложении под заголовком “2.c”.

3. Написать программу, рисующую движущийся символ (при выключенном курсоре, без использования функции стирания экрана).

Для выполнения данного задания, была реализована программа, где вначале идет отключение курсора специальной последовательностью, затем начинается цикл от 1 до 25 для движения по строкам и вложенный в него цикл от 1 до 80 для движения по столбцам, там идет отрисовка символа – 1, ожидание 50000us, сдвиг обратно, затирание символа и далее заного по циклу.



Демонстрация работы программы, как видно в конце выполнения, когда символ останавливает свое движение – курсор включается обратно.

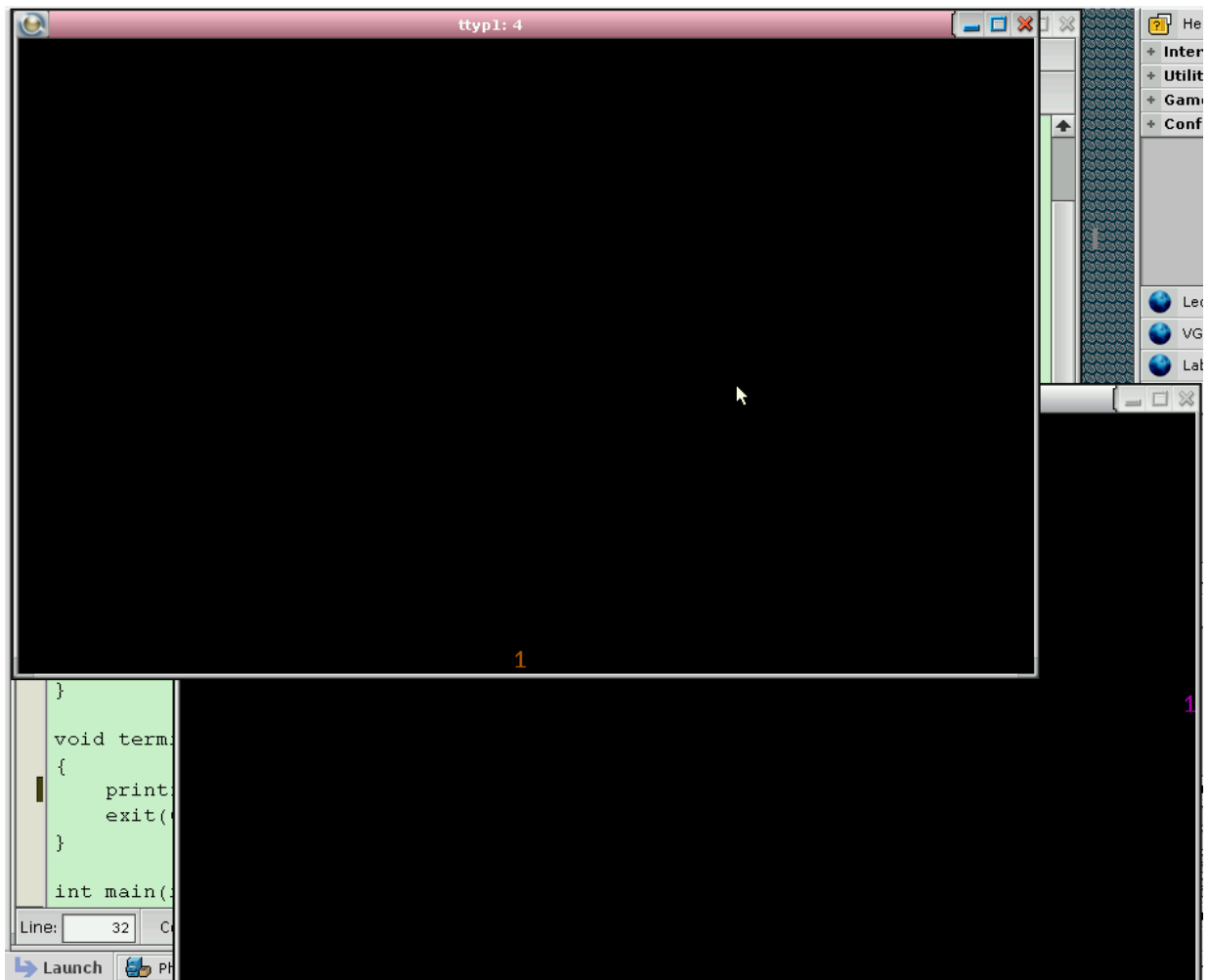
Код программы предоставлен в приложении под пунктом 3.с.

- 4. Написать программу, рисующую бесконечно движущийся символ. Характер движения (скорость, направление, цвет и т.д.) задавать с помощью параметров командной строки. Предусмотреть восстановление параметров дисплея (цвет, курсор) при принудительном завершении программы. Осуществить запуск нескольких экземпляров программы с разными параметрами движения (запуск с одного терминала, вывод на другой).**

Для выполнения данного задания была написана программа, в которой сначала идет считывание аргументов запуска – скорости, направления и цвета символа. Если количество введенных аргументов не 3, то программа завершается ошибкой с кодом -1, если аргументы не входят в установленный диапазон – от 1 и до «бесконечности» для скорости, от 1 до 4 для направления и от 0 до 7 для цвета. Далее идет передача данных параметров в функцию отрисовки символа – `print_symbol(...)`; - в которой идет определение направления и начинается отрисовка символа «1» из середины экрана с определенным цветом и в определенном направлении. Цикл идет бесконечный, в случае аварийного завершения программы выполнится восстановление терминала – возврат курсора, цвета. Чтобы передать программу в другой терминал воспользуемся спец. Символом “>” после которого указывает куда хотим передать выполнения программы, например в терминал 2 - `/dev/tty2`. На скриншоте ниже программа была запущена в двух терминалах с разными цветами и направлением:

```
./4 1 1 4 > /dev/tty1 & ./4 1 4 5 > /dev/tty2
```

Код программы предоставлен в приложении под пунктом 4.с.



ПРИЛОЖЕНИЕ:

Листинг программ:

1.c:

```
#include <stdio.h>  
  
int main()  
{  
  
    printf("\033[12;37HHello\n");  
    return 0;  
}
```

2.c:

```
#include <stdio.h>  
#include </root/raw.h>  
  
int main()  
{  
    printf("\033[=2G\033[=4F");  
}
```

```

        setraw();
        char c = '1';
        while(c != '\n')
        {
            c = getchar();
            printf("Written symbol=%c, code=%d\n\r", c, c-'0');
        }
        unsetraw();
        printf("\033[=0G\033[=7F\n");
        return 0;
    }
}

```

3.c:

```

#include <stdio.h>
#include <unistd.h>
#include </root/raw.h>

int main()
{
    int i, j;
    printf("\033[?25l");
    for(i = 1; i <= 25; i++)
    {
        for(j = 1; j <= 80; j++)
        {
            printf("1");
            fflush(stdout);
            usleep(50000);
            printf("\033[%d;%dH ", i, j);
        }
        printf("\033[?25h");
        return 0;
    }
}

```

4.c:

```

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>

void print_symbol(int speed, int direction, int color)
{
    int dx = 0, dy = 0;
    if (direction == 1)
        dy = 1;
    else if (direction == 2)
        dy = -1;
    else if (direction == 3)
        dx = -1;
    else
        dx = 1;
    setbuf(stdout, NULL);
    printf("\033[2J");
    useconds_t sleep = 50000000 / speed;
    printf("\033[?25l\033[3%dm", color);
    for (int y = 12, x = 40; ; y += dy, x += dx)
    {
        printf("\033[%d;%dH1", y, x);
        usleep(speed);
        printf("\033[D ");
    }
}
}

```

```

void terminal_recovery(int sig)
{
    printf("\033[?25h\033[0m\n");
    exit(0);
}

int main(int argc, char* argv[])
{
    signal(SIGINT, terminal_recovery);
    if (argc != 4)
    {
        printf("Incorrect number of arguments\n");
        exit(-1);
    }
    int speed = atoi(argv[1]);
    int direction = atoi(argv[2]);
    int color = atoi(argv[3]);

    if(direction > 4 || direction < 1 || color > 7 || color < 0 || speed <
1)
    {
        printf("Error in arguments\n");
        exit(-1);
    }

    printf("Speed=%d, direction=%d, color=%d\n", speed, direction, color);
    sleep(2);
    print_symbol(speed, direction, color);
    return 0;
}

```