ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 2

Выполнил:

Студент группы: ИП-712

Алексеев С.В.

Проверил: профессор кафедры ПМиК

Фионов А.Н.

Новосибирск – 2020

## Содержание

[1. Задание 3](#_Toc6575)

[2. Лабораторная работа №2 часть 1 4](#_Toc6576)

[3. Лабораторная работа №2 часть 2 7](#_Toc6577)

[4. Лабораторная работа №2 часть 3 9](#_Toc6578)

[5. Лабораторная работа №2 часть 4 12](#_Toc6579)

[6. Лабораторная работа №2 часть 5 14](#_Toc6580)

# Задание

**LAB 2 Процессы и асинхронное взаимодействие:**

1. Тщательно изучить библиотеку VinGraph.
2. Используя функции библиотеки VinGraph, нарисовать абстрактную картину, которой представлены (почти) все доступные графические элементы.
3. Заставить нарисованные элементы двигаться независимо друг от друга с помощью параллельных процессов (можно изменять во времени положение, цвет, размеры, конфигурацию графических элементов). Предусмотреть завершение программы по нажатию на любую клавишу.
4. Нарисовать нечто, движущееся по замкнутой кривой. Организовать изменение траектории движения по нажатию на клавиши (организуя взаимодействие процессов через общую область памяти (shared memory)). В качестве фона можно использовать (оживленную) картину, созданную на предыдущих этапах работы.
5. Затем последнюю программу сделать с помощью нитей в одном процессе.

# Лабораторная работа №2 часть 1

**Библиотека VinGraph** представляет такое же простое средство для вывода графики, как Borland под MS-DOS, и даже еще более простое, если говорить об анимации созданных изображений.

Вывод графики производится в окне графического терминала VinGraph. Одновременно может быть запущено множество графических терминалов при условии, что каждому из них задается свое уникальное имя. Терминал VinGraph может быть запущен из командной строки, в этом случае он работает, пока пользователь его не закроет. Терминал VinGraph может быть также запущен из программы пользователя, в этом случае он закрывается при (успешном) завершении программы. Программа пользователя, точнее процесс, прикрепляется к какому-то одному терминалу и работает с ним. Несколько процессов могут одновременно выводить графику на один терминал. Один процесс не может делать вывод сразу на несколько терминалов, да это и не нужно.

Все функции библиотеки VinGraph являются нить-безопасными (thread-safe), т.е. могут свободно вызываться из разных нитей вашего процесса. В архитектурном плане сам терминал VinGraph состоит из двух нитей: одна служит для взаимодействия со средой Photon, другая взаимодействует с прикладными программами посредством интерфейса функций библиотеки VinGraph. Пользователь вызывает библиотечную функцию, и та посылает соответствующее сообщение графическому терминалу (и принимает от него ответ).

**Подключение библиотеки к программе:**

В начало текста программы нужно включить файл, содержащий объявления прототипов функций и другие нужные вещи:

#include <vingraph.h>

Вначале выполнения программы, до вызова графических функций, необходимо присоединиться к терминалу VinGraph, а в конце -- отсоединиться от него.

Функция ConnectGraph() может вызываться с аргументом, задающим имя терминала, если вас не устраивает стандартное имя, или вы хотите работать с несколькими терминалами. Данная функция также запускает терминал VinGraph (с соответствующим именем), если он еще не был запущен. Функция CloseGraph() закрывает соединение с терминалом, а также завершает работу терминала, если он был запущен вашей программой.

**Компиляция программы** осуществляется с помощью команды

cc prog.cpp -l vg

Здесь опция l указывает файл библиотеки VinGraph (libvg.a). Теперь вы можете запускать вашу программу:

./a.out

Чтобы запустить терминал VinGraph из командной строки, напечатайте

vg

## Функции рисования

Библиотека VinGraph позволяет выводить на экран следующие элементы графики: точки, прямые и ломаные линии, прямоугольники, многоугольники, окружности, эллипсы, дуги, а также сетки, текстовые надписи и изображения. Кроме того, из этих элементов можно составлять новые композитные элементы (рисунки).

## Установка цвета

Цвет задается числом типа int, в котором три младших байта кодируют интенсивность трех основных цветов -- красного (r), зеленого (g) и синего (b). Для сведения всех трех составляющих цвета в одно целое число используется макрос RGB(r, g, b), определенный в файле vingraph.h. Так, черному цвету соответствует RGB(0, 0, 0), белому -- RGB(255, 255, 255), а промежуточные значения соответствуют всем остальным цветам.

## Управление графическими элементами и создание рисунков

Все графические элементы, созданные с помощью функций рисования, являются объектами, т.е. помнят свое состояние и могут выполнять определенные действия. Каждый графический элемент имеет идентификатор -- положительное целое число. Этот идентификатор возвращается функцией рисования.

## Анимация

Библиотека VingGaph позволяет изменять такие параметры отображаемых элементов, как цвет, местоположение, размер и др. Это позволяет строить изображения, динамически меняющиеся по алгоритму пользователя. **Определение текущих параметров графических элементов**

Параметры всех графических элементов и самого графического терминала находятся под контролем программы пользователя. Но все же в некоторых случаях, особенно при работе множества нитей, бывает удобно запросить информацию о состоянии непосредственно у графического элемента. Такая информация может также понадобиться при отладке программы.

## Ввод с клавиатуры

Для ввода символа с клавиатуры, когда активно окно терминала VinGraph, используется функция char InputChar(). Эта функция ждет нажатия на клавишу и возвращает введенный символ. Функция работает в режиме односимвольного ввода без эхо печати (сырой режим в терминологии терминалов UNIX). При нажатии управляющих клавиш формируются нестандартные коды.

## Графический терминал VinGraph

Графический терминал VinGraph -- это приложение, с которым взаимодействуют функции библиотеки VinGraph.

Графический терминал VinGraph запускается автоматически функцией ConnectGraph() с параметрами, принятыми по умолчанию. Он может быть также запущен из командной строки следующего формата:

vg [-n NAME] [-a X Y W H] [-c R G B]

Все аргументы командной строки должны разделяться пробелами.

Квадратные скобки указывают на необязательность соответствующих опций, сами скобки не должны печататься.

Опция -n задает имя терминала.

Опция -a устанавливает координаты левого верхнего угла, ширину и высоту окна терминала.

Опция -c задает цвет фона путем указания трех составляющих цвета.

Одновременно может быть запущено множество терминалов при условии, что им заданы различные имена.

Переменная среды VGOSC определяет режим работы терминала: с использованием буфера внеэкранного контекста или без него. Использование внеэкранного контекста позволяет убирать мерцание при передвижении графических элементов, однако успешность реализации этого режима зависит от графического адаптера и его драйвера. Этот режим также создает дополнительную нагрузку на центральный процессор.

# Лабораторная работа №2 часть 2

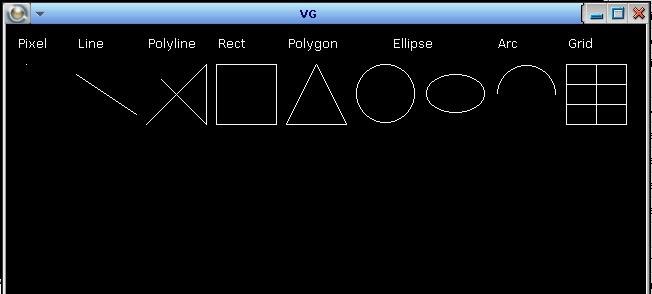


Рисунок 1. Демонстрация работы программы №2

Листинг:

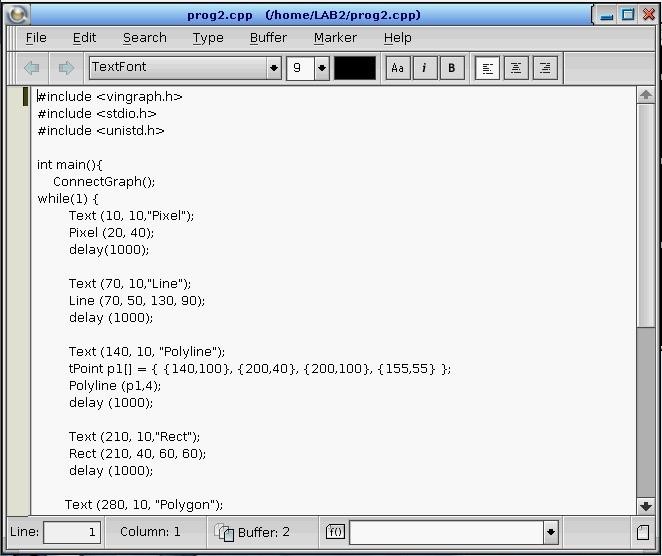


Рисунок 2. Текст программы №2(а)

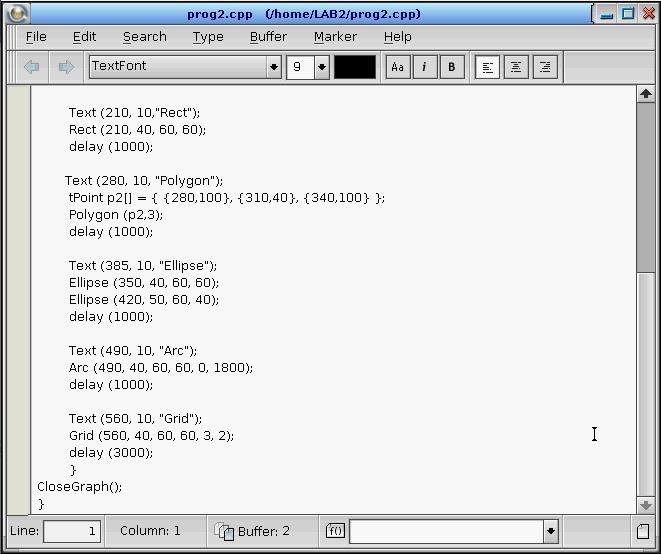


Рисунок 3. Текст программы №2(б)

# Лабораторная работа №2 часть 3

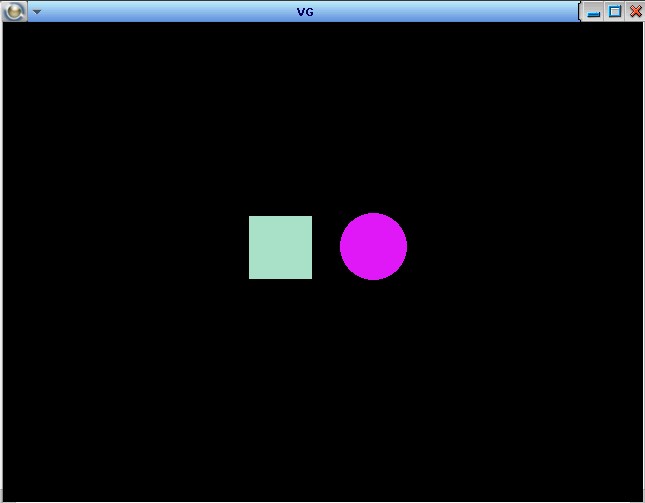


Рисунок 4. Демонстрация работы программы №3(а)

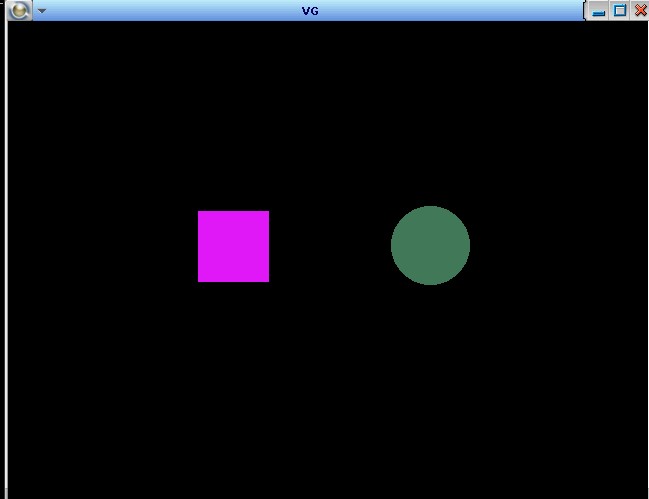


Рисунок 5. Демонстрация работы программы №3(б)

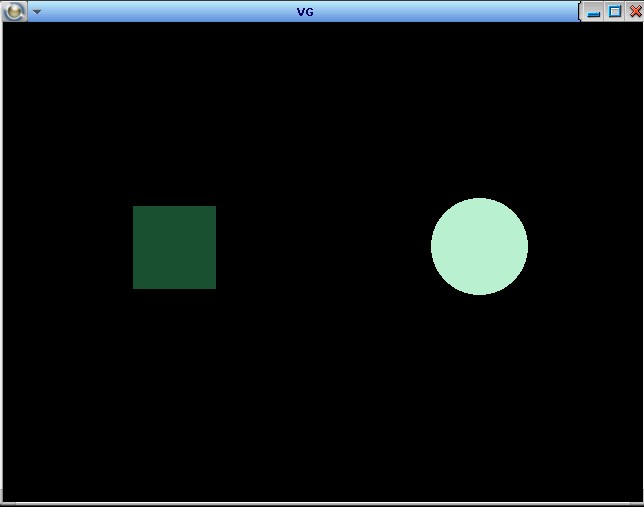
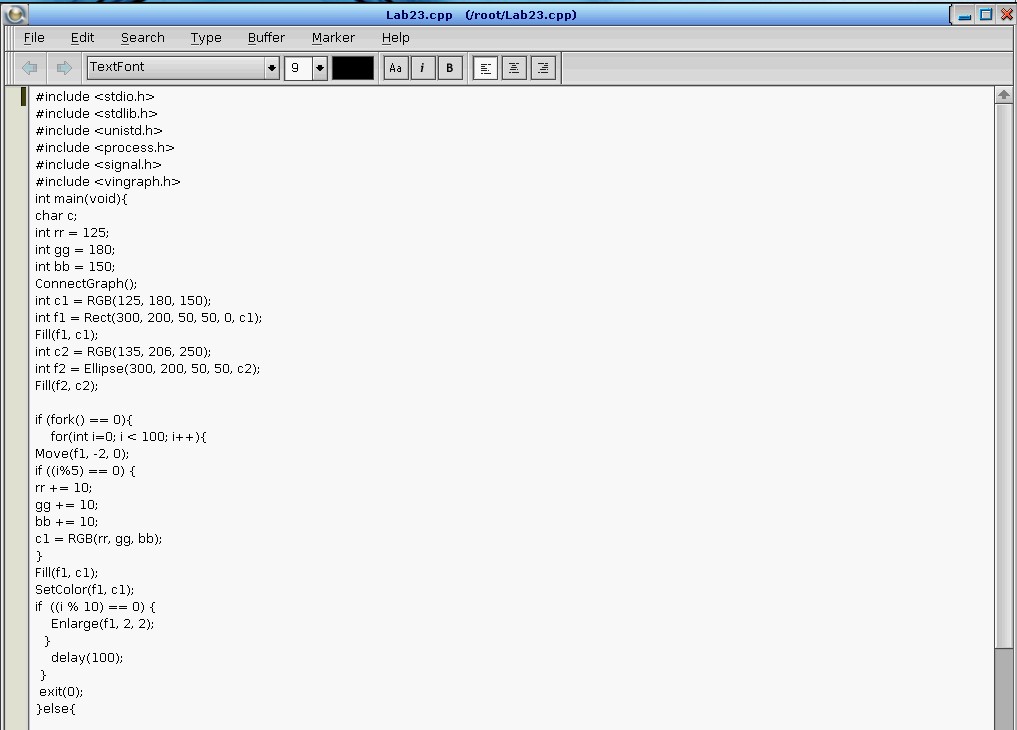


Рисунок 6. Демонстрация работы программы №3(в)



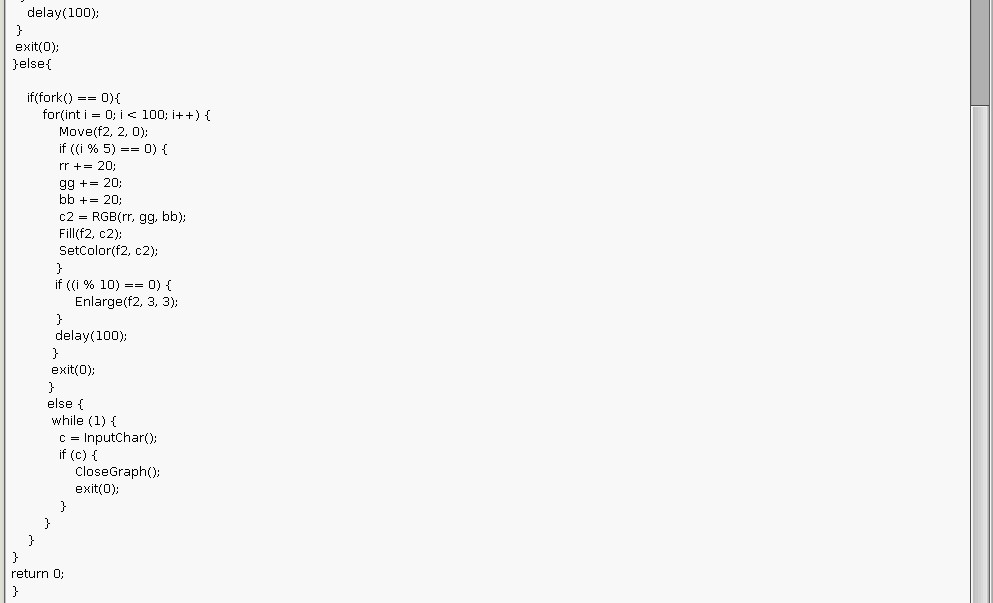


Рисунок 7. Текст программы №3

# Лабораторная работа №2 часть 4

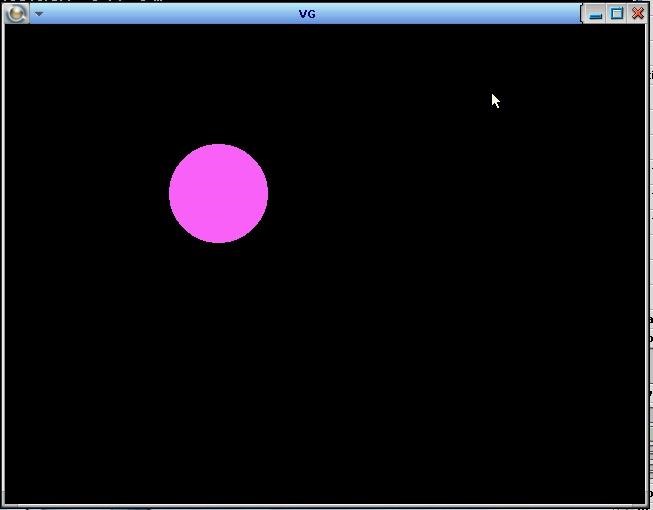


Рисунок 8. Демонстрация работы программы №4(а)

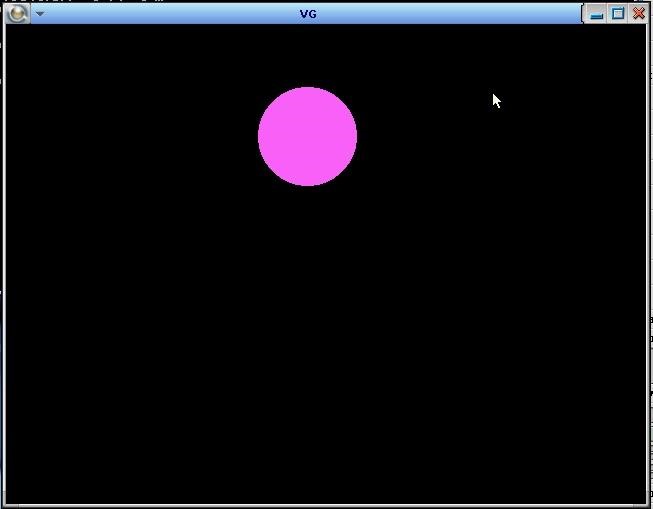


Рисунок 9. Демонстрация работы программы №4(б)

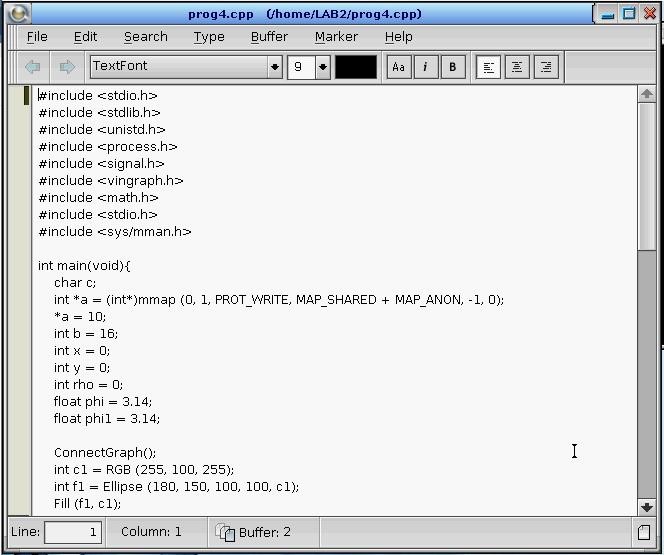


Рисунок 10. Текст программы №4(а)

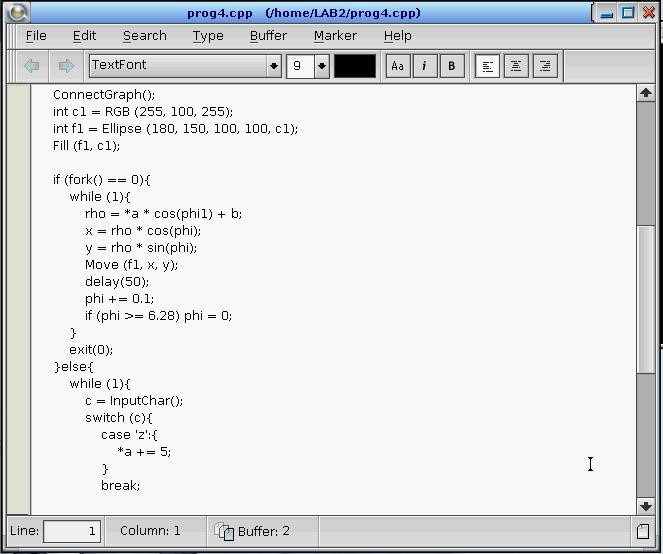


Рисунок 11. Текст программы №4(б)

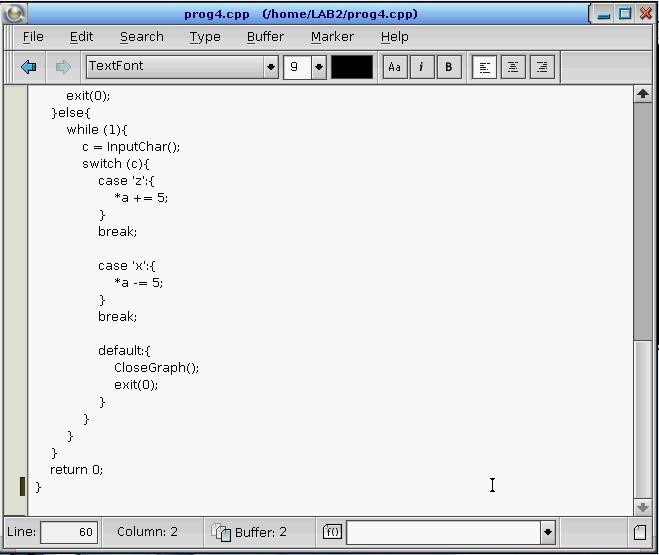


Рисунок 12. Текст программы №4(в)

# Лабораторная работа №2 часть 5

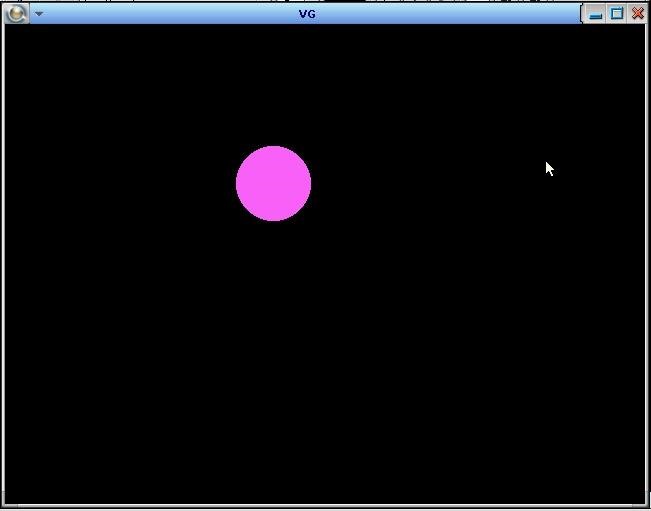


Рисунок 13. Демонстрация работы программы №5

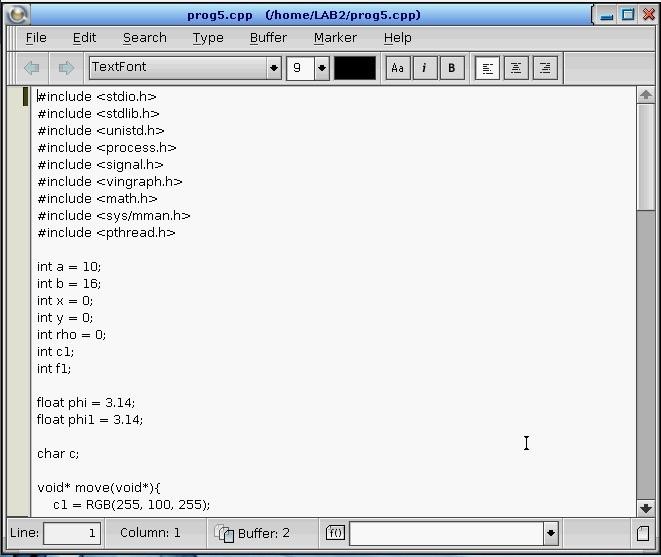


Рисунок 14. Текст программы №5(а)

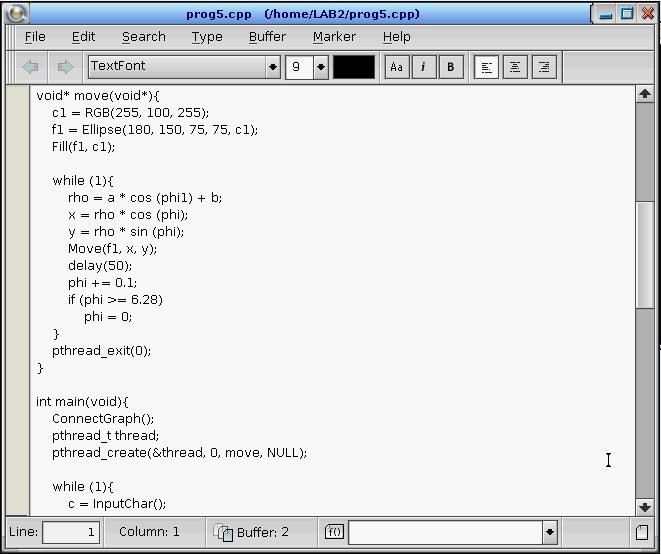


Рисунок 15. Текст программы №5(б)

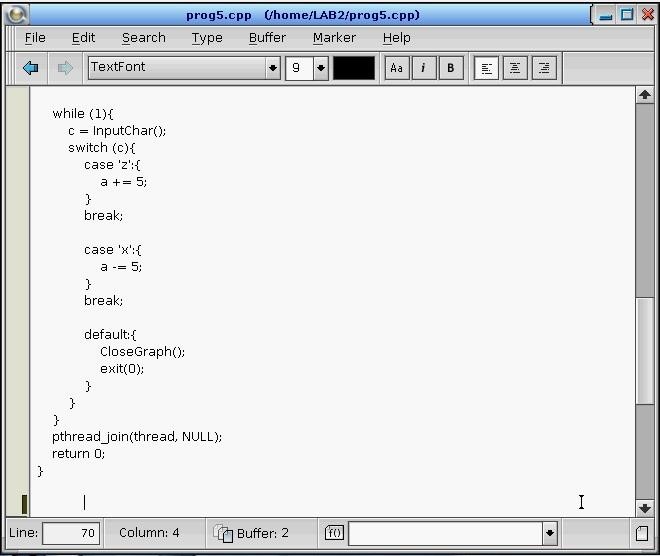


Рисунок 16. Текст программы №5(в)