**Федеральное агентство связи**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Поволжский государственный университет телекоммуникаций и информатики»**

**Факультет информационных систем и технологий**

**Кафедра программного обеспечения и управления в технических системах**

Лабораторная работа №4 Управление вводом и выводом

информации в ОС

Выполнил студент

группы МОИС-01

Богрянцев Игорь

Проверил: к.т.н., доцент

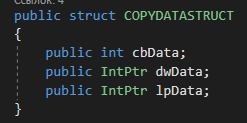
Бадыков Р. И.

Самара, 2022 г.

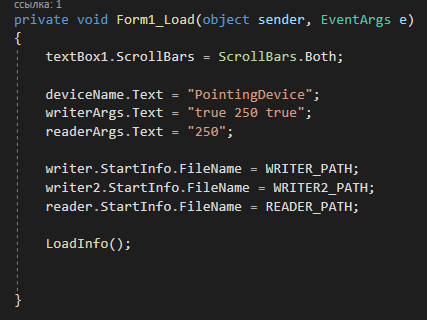
В коде программы , прописываем пути к запускаемым процесса. С помощью функций Process, создаем объекты типа «процессы». Изображение выглядит как текст

Автоматически созданное описание

Используем конструкцию COPYDATASTRUCT, для передачи данных между процессами.

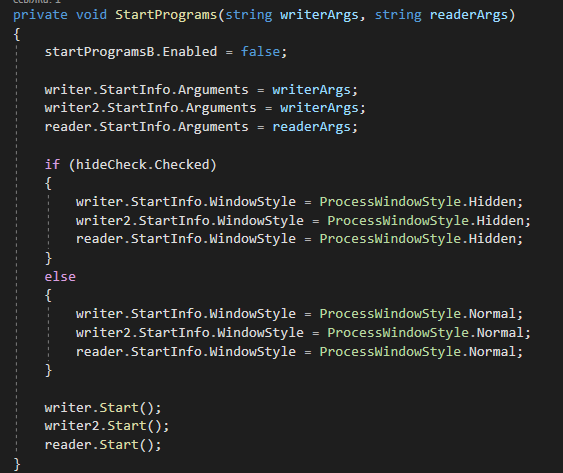


В функцию загрузки формы Form1\_Load(), добавим действия, которые позволяют добавить нам «ключи доступа», аргументы для запуска процессов. deviceName.Text=”PointingDevice”; Для процесса записи используется writerArgs, для процесса чтения используем readerArgs.

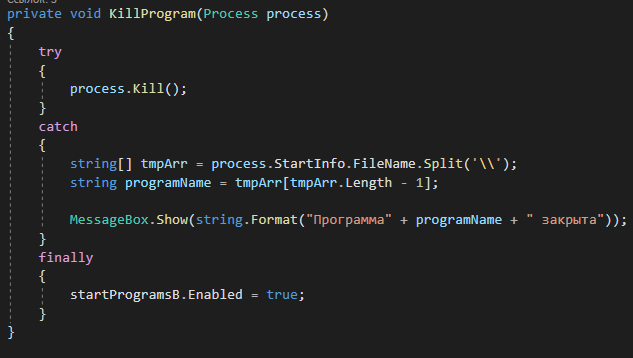


Для запуска программ, мы используем функцию StartPrograms(string writerArgs, string readerArgs), где string writerArg- аргументы для записи, string readerArgs – аргументы для чтения.

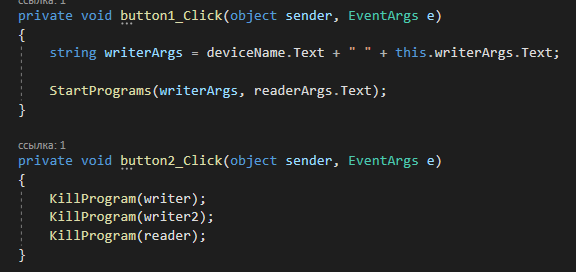
После, происходит проверка на «включение» фонового режима процессов. Если hideChecked присутствует, то потокам задается определенный приоритет. Затем, с помощью метода writer.Start(); - запускается процесс на запись, с помощью reader.Start(); - на чтение.



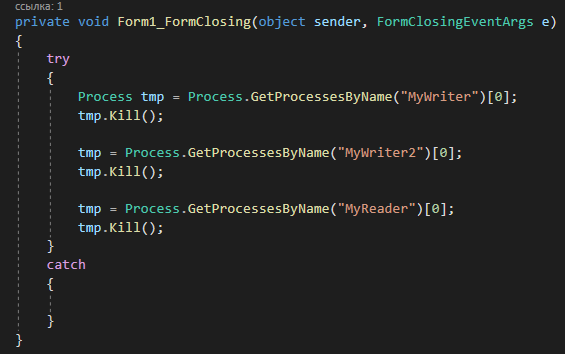
Далее, опишем функцию KillProgram(Process process), в параметры передаются Process process – названия объекта процесса. В методе, происходит проверка на «убийство» процесса и выполняется вывод названия процесса в виде «окна». С помощью строки startProgramsB.Enabled = true; Делаем возможным запуск процессов снова.



Функции нажатий первой и второй кнопок, вызывают запуск и завершение процессов соответственно.



Функция private void Form1\_FormClosing(object sender, FormClosingEventArgs e), используется для завершения процессов с помощью функций закрытия процессов через их названия. Для этого создается объект и ему присваивается имя процесса. Process tmp = Process.GetProcessesByName("MyWriter")[0]; После чего, процесс завершается, благодаря методу .



Далее, была создана функция сохранения информации о девайсе через сохранения таковой в файл. Функция называется saveB\_Click(); В теле метода создается поток записи информации в файл и вызывается метод записи : writer.WriteLine();

Изображение выглядит как текст

Автоматически созданное описание

В событие кнопки «клик», пишем: строку diskName для записи имени диска. С помощью циклов Foreach, получаем cписок накопителей, подключенных через USB и вывод буквы диска.

Изображение выглядит как текст

Автоматически созданное описание



static void EjectDrive(string driveLetter) – функция, которая определяет по букве имени накопителя USB- устройство, позволяющее нам извлечь устройство. IntPtr handle позволяет нам, используя полученную информацию о подключенном устройстве, проверить , возможно ли его отключение. С помощью проверки if ((long)handle == -1), производится решение, ведь если условие совпадает, то нам выводится ошибка.

Изображение выглядит как текст

Автоматически созданное описание

Благодаря методам 

Программа получает контроль для извлечения USB-устройства.