using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace ConsoleApp1

{

public class Program

{

static void Main(string[] args)

{

/\*

https://stackoverflow.com/questions/200469/what-is-the-difference-between-a-process-and-a-thread

\*/

/\*

https://www.geeksforgeeks.org/difference-between-process-and-thread/

\*/

/\*

https://www.guru99.com/difference-between-process-and-thread.html

\*/

//var currentProcess=Process.GetCurrentProcess();

//Console.WriteLine(currentProcess.ProcessName);

//for (int i = 0; i < 10; i++)

//{

//var notePadProcess = Process.Start("notepad.exe");

// Thread.Sleep(100);

//}

//////Thread.Sleep(5000);

//if (Console.ReadKey().Key == ConsoleKey.E)

//{

// var allProcesses = Process.GetProcesses();

// foreach (var p in allProcesses)

// {

// if (p.ProcessName == "notepad")

// {

// p.Kill();

// }

// }

//}

//var search=Console.ReadLine();

//var process = Process.Start($"https://www.youtube.com/results?search\_query={search}");

//var allProcesses = Process.GetProcesses();

//foreach (var p in allProcesses)

//{

// Console.WriteLine(p.ProcessName);

//}

/////////Process.Start("shutdown", "/s /t 0");

//var allProcesses = Process.GetProcesses().Where(p=>p.ProcessName=="notepad").ToList();

//for (int i = 0; i < allProcesses.Count()-1; i++)

//{

// allProcesses[i].Kill();

//}

//Thread thread = new Thread(() =>

//{

// Process.Start("notepad.exe");

//});

//thread.Start();

//Thread thread2 = new Thread(() =>

//{

// Process.Start("calc.exe");

//});

//thread2.Start();

//Thread thread = new Thread(() =>

//{

// for (int i = 0; i < 10000; i++)

// {

// Console.WriteLine("AAAAAAAAAAAA");

// }

//});

//thread.Start();

//thread.Priority= ThreadPriority.Lowest;

//Thread thread2 = new Thread(() =>

//{

// for (int i = 0; i < 10000; i++)

// {

// Console.WriteLine("BBB");

// }

//});

//thread2.Start();

//thread2.Priority= ThreadPriority.Highest;

}

}

}