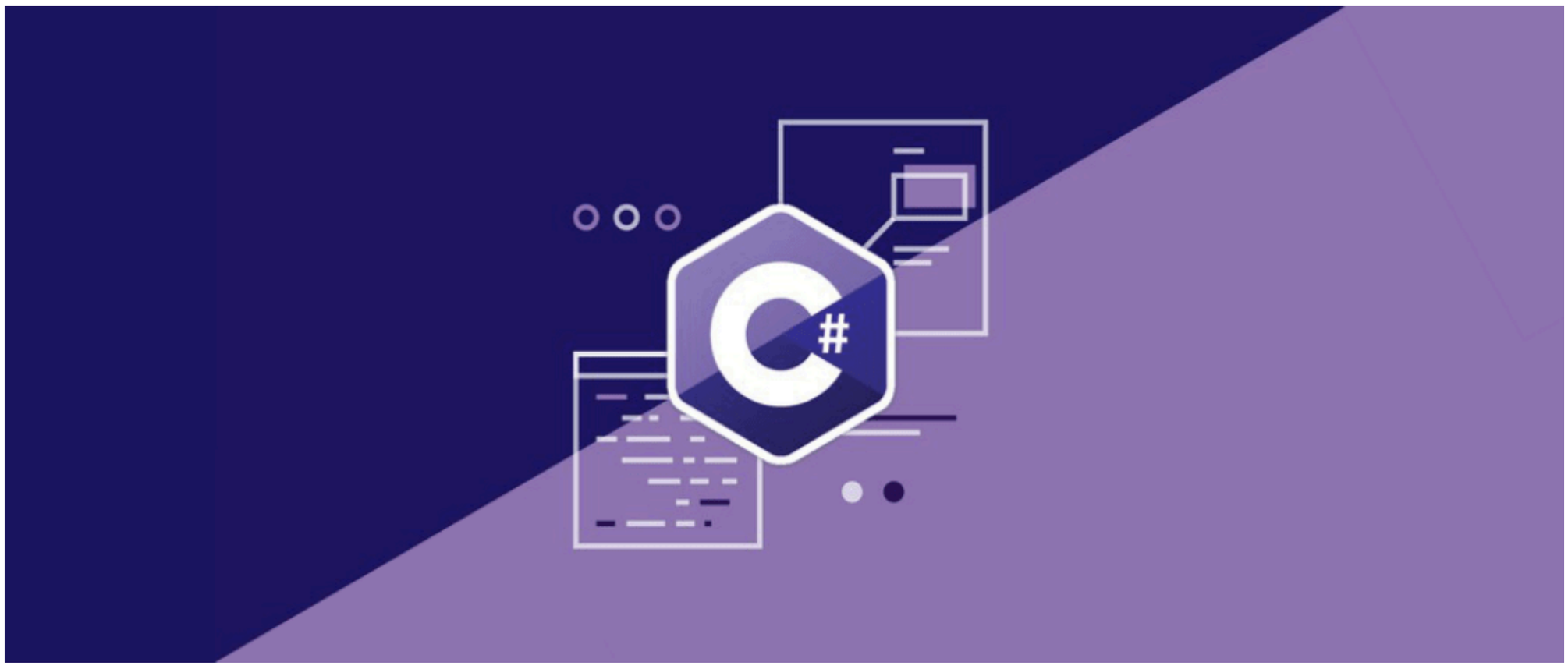
轻松进程通信：C#中使用命名管道实现高效IPC

作者：微信公众号：【架构师老卢】 12-25 18:51



概述：在C#中进行进程间通信，最佳选择是使用命名管道（Named Pipes）。命名管道是一种高性能、轻量且可靠的IPC机制，适用于同一台计算机上的进程通信。

在C#中进行进程间通信（IPC），最佳选择通常是使用命名管道（Named Pipes）或进程间通信（IPC）的其他机制，例如.NET Remoting、WCF、共享内存等。下面分别介绍这些方法，并推荐使用命名管道作为一种轻量、可靠的IPC方法。

1. 命名管道（Named Pipes）：

命名管道是一种高性能、低延迟的IPC机制，适用于同一台计算机上的进程通信。以下是一个简单的例子，演示了两个C#进程之间的通信：

```
using System;
using System.IO;
using System.IO.Pipes;
using System.Text;
using System.Threading.Tasks;

class Program
{
    static void Main()
    {
        // 创建命名管道服务器
        Task.Run(() => StartPipeServer());

        // 创建命名管道客户端
        StartPipeClient();

        Console.ReadLine();
    }

    static void StartPipeServer()
    {
        using (NamedPipeServerStream pipeServer = new NamedPipeServerStream("MyPipe"))
        {
            Console.WriteLine("等待客户端连接...");
            pipeServer.WaitForConnection();

            Console.WriteLine("与客户端建立连接。");

            using (StreamReader reader = new StreamReader(pipeServer))
            {
                string message = reader.ReadToEnd();
                Console.WriteLine($"接收到消息：{message}");
            }
        }
    }

    static void StartPipeClient()
    {
        using (NamedPipeClientStream pipeClient = new NamedPipeClientStream(".", "MyPipe", PipeDirection.Out))
        {
            Console.WriteLine("尝试连接到服务器...");
            pipeClient.Connect();

            Console.WriteLine("已连接到服务器。");

            using (StreamWriter writer = new StreamWriter(pipeClient))
            {
                string message = "Hello, Server!";
                writer.Write(message);
                Console.WriteLine($"发送消息：{message}");
            }
        }
    }
}
```

推荐选择：

命名管道是一种轻量、简单且可靠的IPC机制，适用于同一台计算机上的进程通信。在不需要跨计算机通信的情况下，推荐使用命名管道作为最佳选择。如果需要跨计算机通信，可以考虑使用更复杂的机制，如.NET Remoting、WCF等。