进程（Process）是Windows系统中的一个基本概念，它包含着一个运行程序所需要的资源。进程之间是相对独立的，一个进程无法直接访问另一个进程的数据

**Process 的属性与方法**

System.Diagnostics 命名空间当中存在Process类  
表1.0 显示了Process类的常用属性：

|  |  |
| --- | --- |
| 属性 | 说明 |
| BasePriority | 获取关联进程的基本优先级。 |
| ExitCode | 获取关联进程终止时指定的值。 |
| ExitTime | 获取关联进程退出的时间。 |
| Handle | 返回关联进程的本机句柄。 |
| HandleCount | 获取由进程打开的句柄数。 |
| Id | 获取关联进程的唯一标识符。 |
| MachineName | 获取关联进程正在其上运行的计算机的名称。 |
| MainModule | 获取关联进程的主模块。 |
| Modules | 获取已由关联进程加载的模块。 |
| ProcessName | 获取该进程的名称。 |
| StartTime | 获取关联进程启动的时间。 |
| Threads | 获取在关联进程中运行的一组线程 |

 表1.0

除了上述属性，Process类也定义了下列经常使用的方法：

|  |  |
| --- | --- |
| 方法 | 说明 |
| GetProcessById | 创建新的 Process 组件，并将其与您指定的现有进程资源关联。 |
| GetProcessByName | 创建多个新的 Process 组件，并将其与您指定的现有进程资源关联。 |
| GetCurrentProcess | 获取新的 Process 组件并将其与当前活动的进程关联。 |
| GetProcesses | 获取本地计算机上正在运行的每一个进程列表。 |
| Start | 启动一个进程。 |
| Kill | 立即停止关联的进程。 |
| Close | 释放与此组件关联的所有资源。 |
| WaitForExit | 指示 Process 组件无限期地等待关联进程退出。 |

 表1.1

Process类的详细信息可以参考 [http://msdn.microsoft.com/zh-cn/library/system.diagnostics.process.aspx](http://msdn.microsoft.com/zh-cn/library/system.diagnostics.process.aspx" \t "https://www.cnblogs.com/leslies2/archive/2012/03/06/_blank)  
下面将举例介绍一下Process的使用方式

**建立与销毁进程**

利用 Start 与Kill 方法可以简单建立或者销毁进程

1 static void Main(string[] args)  
2 {  
3 Process process = Process.Start("notepad.exe","File.txt");  
4 Thread.Sleep(2000);  
5 process.Kill();  
6 }

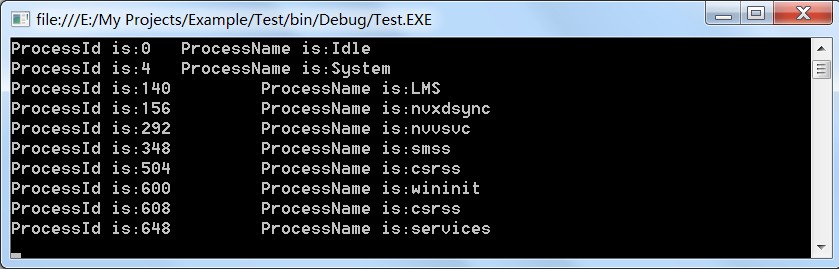
**列举计算机运行中的进程**

使用 GetProcesses 方法可以获取本地计算机上正在运行的每一个进程列表。

1 static void Main(string[] args)  
 2 {  
 3 var processList = Process.GetProcesses()  
 4 .OrderBy(x=>x.Id)  
 5 .Take(10);  
 6 foreach (var process in processList)  
 7 Console.WriteLine(string.Format("ProcessId is:{0} \t ProcessName is:{1}",  
 8 process.Id, process.ProcessName));  
 9 Console.ReadKey();  
10 }

[IMG_257](https://www.cnblogs.com/leslies2/archive/2012/03/06/javascript:void(0);)

运行结果



如果已知进程的Id，就可以通过 GetProcessById 方法获取对应的进程。  
同样地，你也可能通过GetProcessByName方法获取多个对应名称的进程。

**获取进程中的多个模块**

Process类的Modules属性，通过此属性可能获取进程中的多个模块。  
这些模块可以是以 \*.dll 结尾的程序集，也可以是 \*.exe 结尾的可执行程序。

1 static void Main(string[] args)  
2 {  
3 var moduleList = Process.GetCurrentProcess().Modules;  
4 foreach (System.Diagnostics.ProcessModule module in moduleList)  
5 Console.WriteLine(string.Format("{0}\n URL:{1}\n Version:{2}",  
6 module.ModuleName,module.FileName,module.FileVersionInfo.FileVersion));  
7 Console.ReadKey();  
8 }

[IMG_262](https://www.cnblogs.com/leslies2/archive/2012/03/06/javascript:void(0);)

运行结果：

