


```
semaphore.Release();  
...
```

```
##### **4.2 ##### **
```

```
- **##### **
```

```
```csharp
```

```
// #####
```

```
var pipeSecurity = new PipeSecurity();
```

```
pipeSecurity.AddAccessRule(new PipeAccessRule("Everyone", PipeAccessRights.ReadWrite,
AccessControlType.Allow));
```

```
var pipeServer = new NamedPipeServerStream("SecurePipe", PipeDirection.InOut, 1,
PipeTransmissionMode.Byte, pipeSecurity);
```

```
...
```

```

```

```
5. **##### **
```

```
**5.1 ##### **
```

```
```csharp
```

```
// #####
```

```
Process.GetCurrentProcess().PriorityClass = ProcessPriorityClass.High;
```

```
// #####
```

```
foreach (Process process in Process.GetProcesses())
```

```
{  
    Console.WriteLine($"PID: {process.Id}, Name: {process.ProcessName}, Priority:  
{process.PriorityClass}");
```

```
}
```

```
...
```

```
##### **5.2 ##### **
```

```
- **##### **
```

```
```csharp
```

```
Process targetProcess = Process.Start("notepad.exe");
```

```
targetProcess.EnableDebugging = true;
```

```
...
```

```
- **##### **
```

```
```csharp
```

```
// P/Invoke Windows API DLL
```

```
[DllImport("kernel32.dll")]
```

```
static extern IntPtr OpenProcess(uint dwDesiredAccess, bool bInheritHandle, int dwProcessId);
```

```
[DllImport("kernel32.dll")]
```

```
static extern IntPtr VirtualAllocEx(IntPtr hProcess, IntPtr lpAddress, uint dwSize, uint  
flAllocationType, uint flProtect);
```


...

```
##### **8.3 [ ] [ ] [ ] [ ] [ ] [ ] **
```csharp
// [] [] [] [] CPU [] [] [] [] [] []
private void MonitorProcessResources()
{
 Process target = Process.GetProcessById(1234);
 while (!target.HasExited)
 {
 Console.WriteLine($"CPU: {target.TotalProcessorTime}, Memory: {target.WorkingSet64}
bytes");
 Thread.Sleep(1000);
 }
}
```
```

```
### 9. ** [ ] [ ] [ ] [ ] [ ] [ ] **
##### **9.1 [ ] [ ] [ ] [ ] [ ] [ ] **
- ** [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] IPC [ ] [ ] [ ] [ ] [ ] [ ]
- ** [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
```csharp
// [] [] [] [] [] []
if (WaitHandle.WaitAny(new[] { mutex, timeoutEvent }) == 0)
{
 // [] [] [] [] [] []
}
```
```

```
##### **9.2 [ ] [ ] [ ] [ ] [ ] [ ] **
- ** [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] `System.Diagnostics`** [ ]
```csharp
var performanceCounter = new PerformanceCounter("Process", "Working Set - Private",
Process.GetCurrentProcess().ProcessName);
Console.WriteLine($" [] [] [] [] [] [] {performanceCounter.NextValue()} bytes");
```
```

```
##### **9.3 [ ] [ ] [ ] [ ] [ ] [ ] **
```csharp
// [] [] [] [] [] [] [] [] System.Diagnostics.Process []
Process.Start(new ProcessStartInfo("ls", "-l") { UseShellExecute = false });
```
```
