





class Program

{

/\* coder:释迦苦僧 \*/

static void Main(string[] args)

{

/\* 创建一个任务 不调用 不执行 状态为Created \*/

Task tk = new Task(() =>

{

});

Console.WriteLine(tk.Status.ToString());

/\* 创建一个任务 执行 状态为 WaitingToRun \*/

Task tk1 = new Task(() =>

{

});

tk1.Start();/\*对于安排好的任务，就算调用Start方法也不会立马启动 此时任务的状态为WaitingToRun\*/

Console.WriteLine(tk1.Status.ToString());

/\* 创建一个主任务 \*/

Task mainTask = new Task(() =>

{

SpinWait.SpinUntil(() =>

{

return false;

}, 30000);

});

/\* 将子任务加入到主任务完成之后执行 \*/

Task subTask = mainTask.ContinueWith((t1) =>

{

});

/\* 启动主任务 \*/

mainTask.Start();

/\* 此时子任务状态为 WaitingForActivation \*/

Console.WriteLine(subTask.Status.ToString());

/\* 创建一个任务 执行 后 等待一段时间 并行未结束的情况下 状态为 Running \*/

Task tk2 = new Task(() =>

{

SpinWait.SpinUntil(() => false, 30000);

});

tk2.Start();/\*对于安排好的任务，就算调用Start方法也不会立马启动\*/

SpinWait.SpinUntil(() => false, 300);

Console.WriteLine(tk2.Status.ToString());

/\* 创建一个任务 然后取消该任务 状态为Canceled \*/

CancellationTokenSource cts = new CancellationTokenSource();

Task tk3 = new Task(() =>

{

for (int i = 0; i < int.MaxValue; i++)

{

if (!cts.Token.IsCancellationRequested)

{

cts.Token.ThrowIfCancellationRequested();

}

}

}, cts.Token);

tk3.Start();/\*启动任务\*/

SpinWait.SpinUntil(() => false, 100);

cts.Cancel();/\*取消该任务执行 但并非立马取消 所以对于Canceled状态也不会立马生效\*/

SpinWait.SpinUntil(() => false, 1000);

Console.WriteLine(tk3.Status.ToString() + " " + tk3.IsCanceled);

SpinWait.SpinUntil(() => false, 1000);

Console.WriteLine(tk3.Status.ToString() + " " + tk3.IsCanceled);

SpinWait.SpinUntil(() => false, 1000);

Console.WriteLine(tk3.Status.ToString() + " " + tk3.IsCanceled);

/\*创建一个任务 让它成功的运行完成 会得到 RanToCompletion 状态\*/

Task tk4 = new Task(() =>

{

SpinWait.SpinUntil(() => false, 10);

});

tk4.Start();

SpinWait.SpinUntil(() => false, 300);

Console.WriteLine(tk4.Status.ToString());

/\*创建一个任务 让它运行失败 会得到 Faulted 状态\*/

Task tk5 = new Task(() =>

{

throw new Exception();

});

tk5.Start();

SpinWait.SpinUntil(() => false, 300);

Console.WriteLine(tk5.Status.ToString());

Console.ReadLine();

}

}