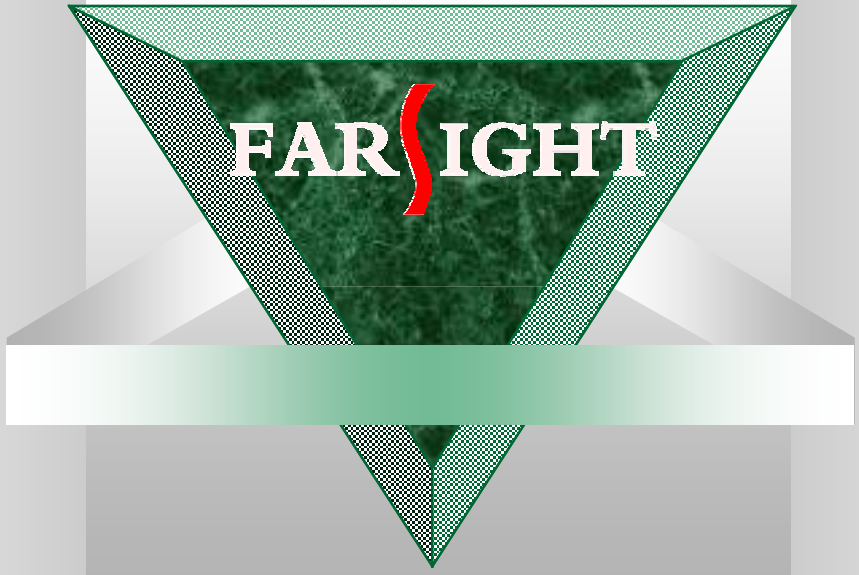


Active Object技术讲解

沈青海

大纲

- AO概述
- 活动规划器类CActiveScheduler
- 活动对象基类CActive
- 常见的活动对象错误
- 总结及特别提示

FARSIGHT

(1)异步系统

- 在现代计算中,异步系统非常流行,存在大量可用于实现异步的方式,其中
  - 在抢占系统中使用多线程
  - 协作式多任务

(2)Active Object (AO)框架

- AO 框架是运行于一个线程内部的调度框架。
- 其基本思想就是把一个单线程分为多个时间片，来运行不同的任务。

FARSIGHT

(3)多线程和AO框架的比较

多线程

- 可以被抢占
- 上下文切换耗费CPU时间
- 由操作系统调度
- 每个线程都有至少4K Stack
- 操作系统还要分配额外的资源记录线程

活动对象

- 不可被抢占
- 没有上下文切换
- 由线程自己的AO框架调度
- AO没有单独的Stack
- 只是一个 Active Object

FARSIGHT

(4)Active Object (AO)框架

- AO 框架是运行于一个线程内部的调度框架。
- 基本思想是把一个单线程分为多个时间片，来运行不同的任务。
- AO 框架包括 CActiveScheduler 和 CActive (Active Object)
- 一个线程的所有的 Active Object 都被安装在 CActiveScheduler.
- CActiveScheduler监控每个 Active Object 是否完成了当前任务

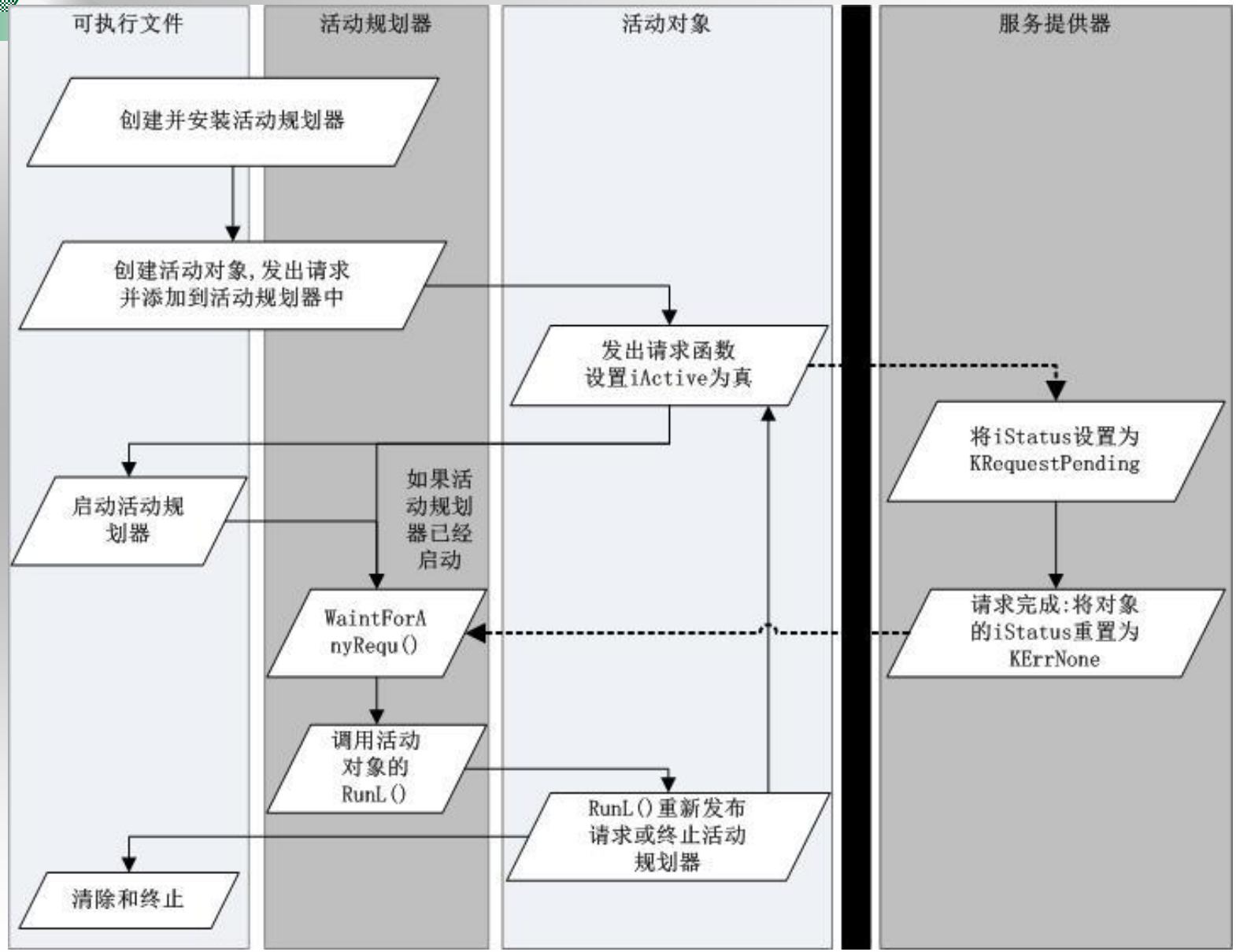
FARSIGHT

大纲

- AO概述
- 活动规划器类CActiveScheduler
- 活动对象基类CActive
- 常见的活动对象错误
- 总结及特别提示

FARSIGHT

(1)活动规划器的生命周期



(2)活动规划器类CActiveScheduler

(1)创建规划器:

```
CActiveScheduler * scheduler = new (ELeave) CActiveScheduler;
CleanupStack:: PushL(scheduler);
CActiveScheduler::Install(scheduler);
```

(2)运行规划器:

```
CActiveScheduler::Start();
```

(3)停止规划器:

```
CActiveScheduler::Stop();
```

FARSIGHT

大纲

- AO概述
- 活动规划器类CActiveScheduler
- 活动对象基类CActive
- 常见的活动对象错误
- 总结及特别提示

FARSIGHT

(1)活动对象使用二阶段构造

```
CExampleLoader:: CExampleLoader(RFs& aFs , CElementList& aElementList ,
        MCsvFileLoaderObserver& aObserver) : CActive (EPriorityStandard)
{
}
void CExampleLoader ::ConstructL (const TDesC& aFileName)
{
    iFileName = aFileName;
    User::LeaveIfError(iFile.open(iFs , iFileName , EFileRead));
    User::LeaveIfError(iTimer.CreateLocal());
    CActiveScheduler::Add(this);
}
```

通过句柄RTimer和RFile链接到所需的两个异步服务器。

必须将活动对象都添加到活动规划器中，并且只能添加一次。

FARSIGHT

(2)启动活动对象

```
void CExampleLoader ::Start( )
{
    TInt delay = (iFileName.Size() % 10) * 100000;
    iTimer.After(iStatus , delay);
    SetActive();//将iActive设为Etrue
}
```

异步服务提供器（运行在另一线程或进程中）将通过完成下面两件事来用信号通知线程（活动对象所属的线程）。

（1） 增加线程的信号量（这个信号量，一般情况用不到。只是当活动对象所属的线程被挂起时，通过增加信号量重新唤醒这个线程，而线程的挂起通过函数WaitForAnyRequest（））。

（2） 将给定的TRequestStatus设置为不同于KRequestPending的值（如果一切运行良好，则很有可能是KErrNone）。

FARSIGHT

(3)RunL()

- 该函数将执行大量任务：
  - 决定下一次迭代应该做什么（加载数据或浪费时间）
  - 检查最后一次迭代状态，并且将这个状态报告给观察器
  - 处理所有加载的数据
- 一般活动对象的这个RunL（）方法希望完成上述的一项或几项工作。
- 在这里还可以再次调用Start（），也就是说可以再次发布服务请求。

FARSIGHT

(4)在RunError()中处理错误

- 活动对象完全允许RunL（）异常退出，结尾是“L”表明了这一点。
- 如果该函数确实异常退出，则它异常退出时的错误码将被传递RunError()。

```
TInt CExampleLoader ::RunError(TInt aError)
{
    iObserver.NotifyLoadCompleted(aError , *this);
    return KErrNone;
}
```

这里的错误处理很简单，仅仅将错误传递给了观察器。

FARSIGHT

(5)删除未完成的请求

```
void CExampleLoader ::DoCancel()
{
    if(iTimering || !iHaveTriedToLoad)
    {
        iTimer.Cancel();
    }
}
```

CActive::Cancel调用DoCancel（），不允许重写CActive::Cancel（）自身，因为该函数完成了大量重要的工作。

（1）检查活动对象是否实际上处于活动状态。如果不是，它就会返回，而不作任何事情。

（2）调用DoCancel（）

（3）等待请求完成，这必须尽可能完成（原始请求可能在删除它之前就已经完成）

（4）将iActive设为假。

FARSIGHT

(6)析构函数

```
CExampleLoader ::~ CExampleLoader ()
{
    Cancel();
    iFile.close();
    iTimer.Close();
}
```

任何活动对象析构函数的第一步都是调用Cancel（）删除任何未完成的请求。如果删除一个带有未完成请求的活动对象，则产生一个请求迷失的严重错误。

必须关闭异步服务提供器的任何句柄，从而避免资源泄漏。

FARSIGHT

(7)启动活动规划器

启动活动规划器前，必须完成如下步骤：

（1）实例化活动规划器

（2）将其安装到线程中

（3）创建一个活动对象并添加到活动规划器中

（4）发出一个请求

```
void DoExampleL()
{
    CActiveScheduler* scheduler = new(ELeave) CActiveScheduler;
    CleanupStack::PushL(scheduler);
    CActiveScheduler::Install(scheduler);
    CElementsEngine* elementEngine = CElementsEngine::NewLC(*console);
    elementEngine->LoadFromCsvFilesL();//发出请求
    CActiveScheduler::Start();//启动活动规划器
    CleanupStack::PopAndDestroy(2 , scheduler);
}
```

FARSIGHT

大纲

- AO概述
- 活动规划器类CActiveScheduler
- 活动对象基类CActive
- 常见的活动对象错误
- 总结及特别提示

FARSIGHT

常见的活动对象错误

- 启动活动对象前忘记调用CActiveScheduler::Add()
- 在发布或重新发布异步请求时没有调用SetActive()
- 将相同的iStatus同时传递给两个服务提供器（因此在相同的活动对象上有多个未完成的请求）
- 长时间运行RunL()

FARSIGHT

大纲

- AO概述
- 活动规划器类CActiveScheduler
- 活动对象基类CActive
- 常见的活动对象错误
- 总结及特别提示

FARSIGHT

(1)特别提示

- Symbian仍然属于抢占式多任务操作系统(这意味着每个线程都有自己的执行时间，直到系统将CPU使用权给予其他线程。当系统调度时，具有最高优先权的线程将首先获得执行)。
- Symbian 系统本身使用了大量的 AO 框架来实现一些系统服务。
- Symbian官方推荐使用活动服务对象(CActive)来代替多线程的使用。

FARSIGHT

(2)Symiban培训初级班

- Symbian SDK及其开发环境
- symbian二进制代码的三种类型
- 打包发布工具使用
- 基本类型、描述符和动态数组
- Symbian的异常处理及清理机制
- 文本显示和字体使用
- UI组件的使用
- 活动对象
- 图形显示和交互
- 对话框
- 文件、流和存储
- 网络应用程序开发
- 多媒体应用程序的开发
- 蓝牙应用

FARSIGHT

(3)Symiban培训高级班

- 文件服务器及流处理
- 多媒体，视频播放以及流媒体技术
- 网络连接处理
- 蓝牙处理
- 活动对象
- 客户端-服务器框架
- ECOM
- 消息传送及通话
- 应用程序签名

FARSIGHT

www.farsight.com.cn

谢谢！