

[SCA] 启动框架技术方案设计

[SCA] 启动框架技术方案设计

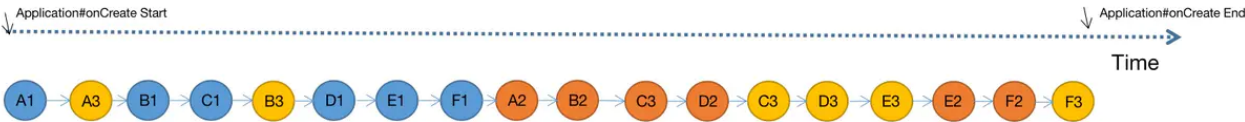
文档历史

版本	修订日期	修订者	修订内容
1.0	2021.4.25	谢敬伟	初稿

1、需求概况

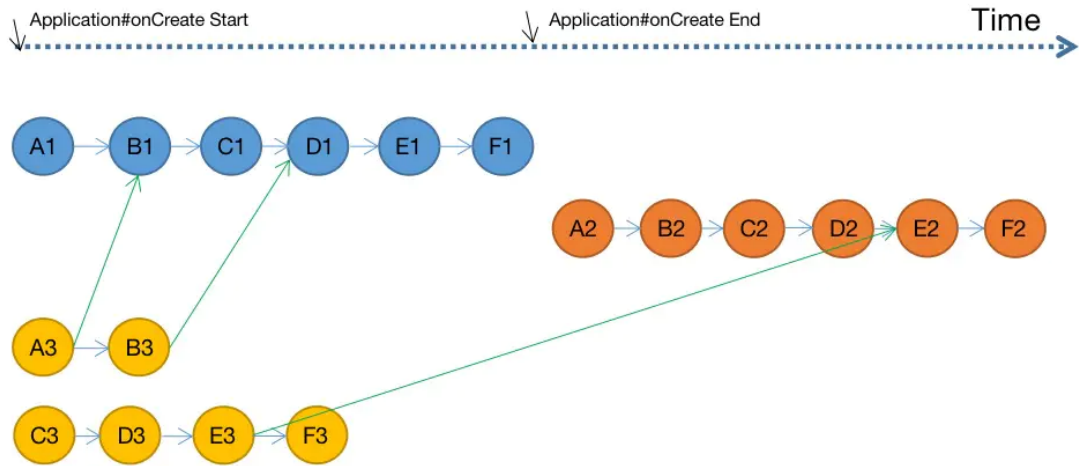
1.1 需求背景

随着移动端应用的功能迭代，项目依赖库越来越多，在应用启动时承载的初始化逻辑变得越来越复杂，影响首帧显示时机。倘若不进行任务分类初始化，则首帧显示前需要完成所有任务的初始化（如下所示）。为了合理安排启动任务，引入启动框架，优化应用启动速度。



1.2 需求目的

根据任务的依赖关系和优先级，合理安排启动任务的执行顺序，提高应用的启动速度，实现效果如下图所示。



1.3 需求功能

基于图结构，支持同异步依赖任务初始化，主要功能包括

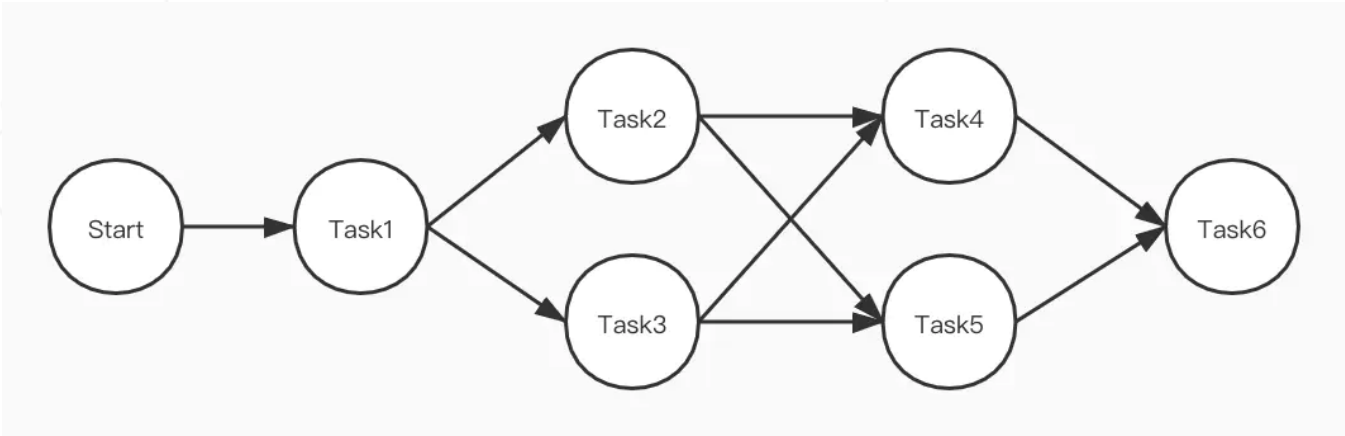
1. 支持为不同进程设置不同的启动模式
2. 支持同步异步任务的调度切换
3. 运行过程动态调整任务执行顺序
4. 支持主动请求阻塞任务

调研文档: <https://docs.google.com/document/d/1W3kOozueSIEaBmIhjrnlbT3mC4FmmtbUncA4V6cusxY/edit>

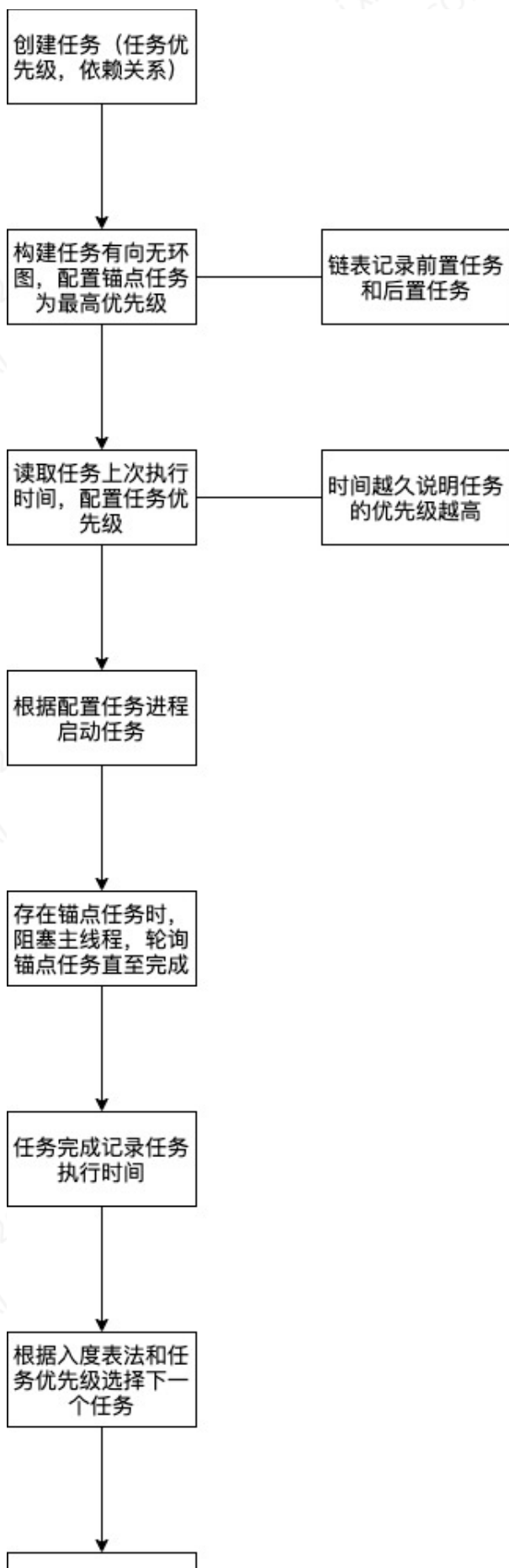
2、技术方案设计

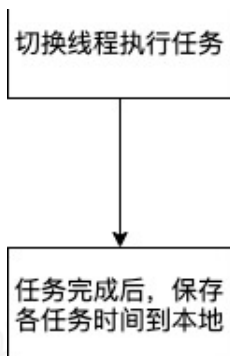
2.1 整体方案设计

方案将主要综合调研方案，结合Anchor和SmartStart，在任务初始阶段构建任务的有向无环图（如下所示），在启动时会记录每个任务时间，动态调整任务执行的顺序。



2.2 详细方案设计





在启动任务时，读取上一次每个任务的执行时间，根据启动任务的依赖关系，使用链表记录前置任务和后置任务，构建任务时根据任务的执行时间设置任务的优先级。

当任务存在锚点任务(锚点任务为阻塞型任务)时，遍历锚点任务的依赖任务，设置锚点任务的优先级，启动时轮询锚点任务，阻塞当前线程，直至锚点任务完成。

任务完成后，通知后置任务，后置任务根据前置任务是否已经全部完成，根据优先级安排任务执行顺序，切换设置线程运行。

当所有任务完成后，记录每个任务的实际执行时间，等待下次启动时使用。

该方案结合了Anchor和SmartStart的优点，相比于Anchor框架来说，由于记录了每次任务执行时间，在下次动态调整任务的执行顺序，能够提高启动速度，相对于SmartStart，增加了锚点，支持阻塞任务，更符合实际应用场景。

2.3 数据存储设计

全部任务启动完成后，记录每个任务的运行时间，由于任务数目较少，且结构比较简单，考虑使用sharepreferences进行存储，每个任务的记录结构如下：

key	type	value	type
task_name	String	time	long

其他

[StartUp](#)

[Alpha](#)

[Anchors](#)

[SmartStart](#)