部件设计

部件设计，主要分析，系统在运行时可能产生的一些非功能性的需求，解决资源抢占造成的冲突，并给出详细的设计方案

主要有4个步骤：

1. 分析并发需求
2. 针对某个需求的设计方案
3. 生命周期
4. 映射到现实系统
5. 分析并发需求

系统在运行时可能因为多人在线或者其他原因产生多cpu，多线程多进程等并发需求。这些需求可以从之前的问题陈述中以及补充规约中获得

类图中并未规定系统运行时功能，从而在系统运行时，可能因为多进程多线程并发需求等情况，进而产生资源抢占等种种问题。部件设计会分析系统运行时产生的一系列非功能性问题并给出解决方案。

本系统在运行时，可能产生的并发需求有：

1. 多人同时使用送水系统并且同时下订单
2. 下达的订单数超过送水工能服务的数量
3. 针对某个需求的设计方案

1中分析了系统可能产生的并发需求，本节就某个需求给出详细的设计方案

A.找出有冲突的相关类 （系统运行时也称为线程进程 ）

B.给出针对相关类的设计方案。如送水系统里同时下达的订单数量不能超过 100单，那么在下订单相关类中可以设计进程池或者通过 if循环控制条件

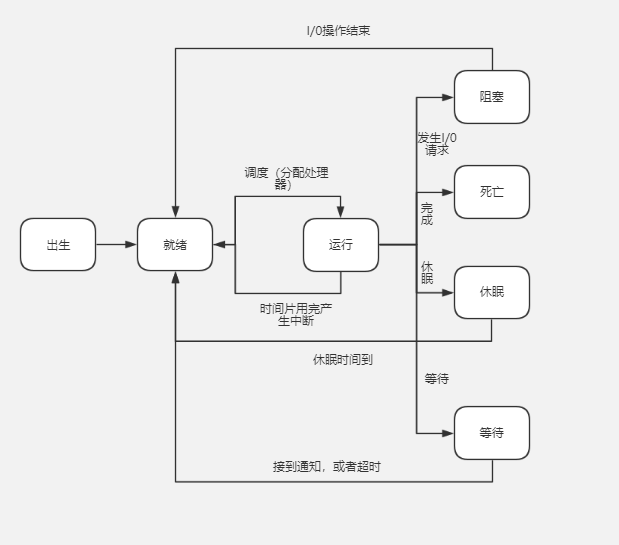
针对订单数量超过送水工总服务能力的问题

在系统运行的时候，同时需要服务的订单太多，可能会出现让送水工忙不过来，此时就发生了送水工人力资源冲突的问题。为了解决这个问题，所有订单按接收的时间顺序安排一个优先级，越早下达的订单，优先级越高，优先级越高的，就越先出现在送水工的服务界面上，让送水工去完成，从而根据先后顺序，合理安排送水工的人力资源。

1. 生命周期

主要描述设计方案中涉及的相关类的生命周期，如线程在什么时候创建，在什么时候结束等。

线程图如下：



1. 映射到现实系统

之前的分析，给出了问题的详细设计方案，但是在现实系统中还需要考虑一些其他问题，比如同学们下订单使用的移动设备以及送水工接收订单使用的移动设备，还有网络因素等等情况。这一节中需要分析以及说明设计方案是否能够在现实系统中运行