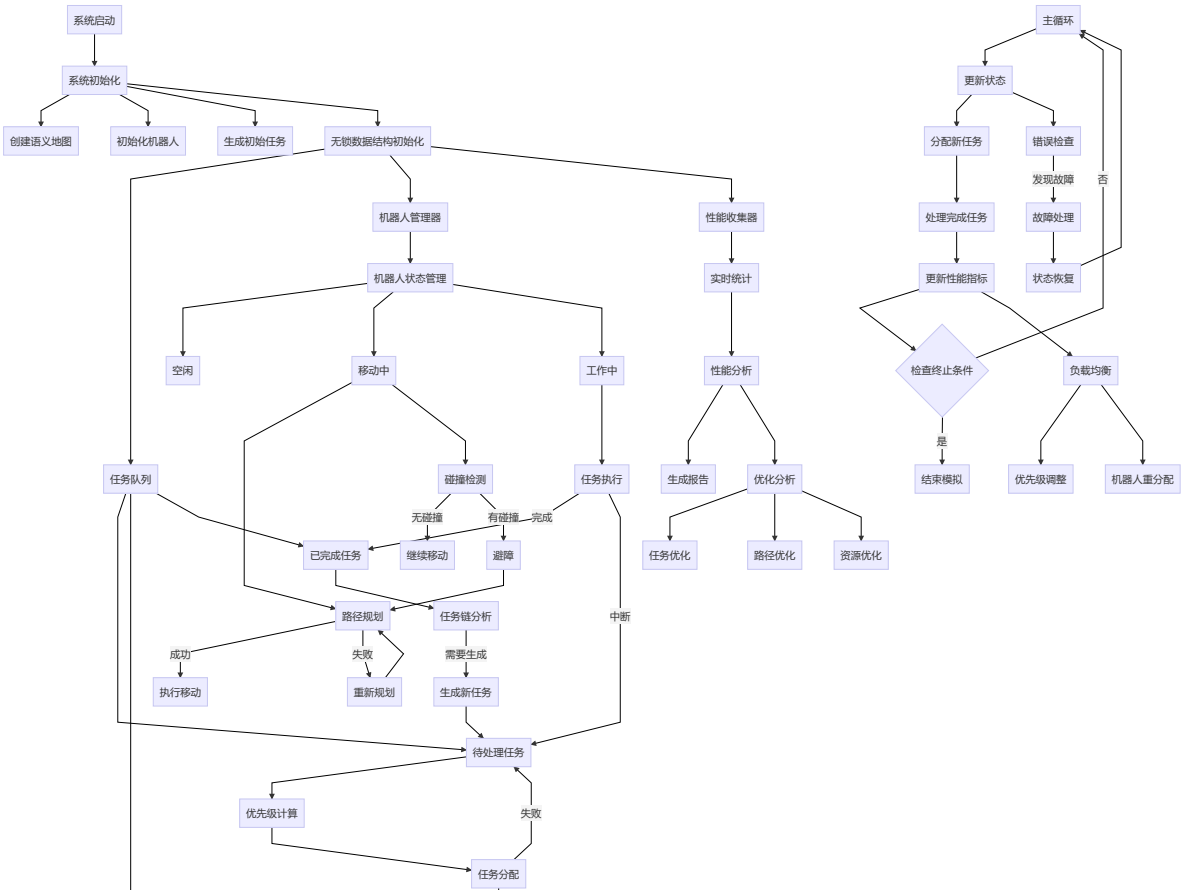
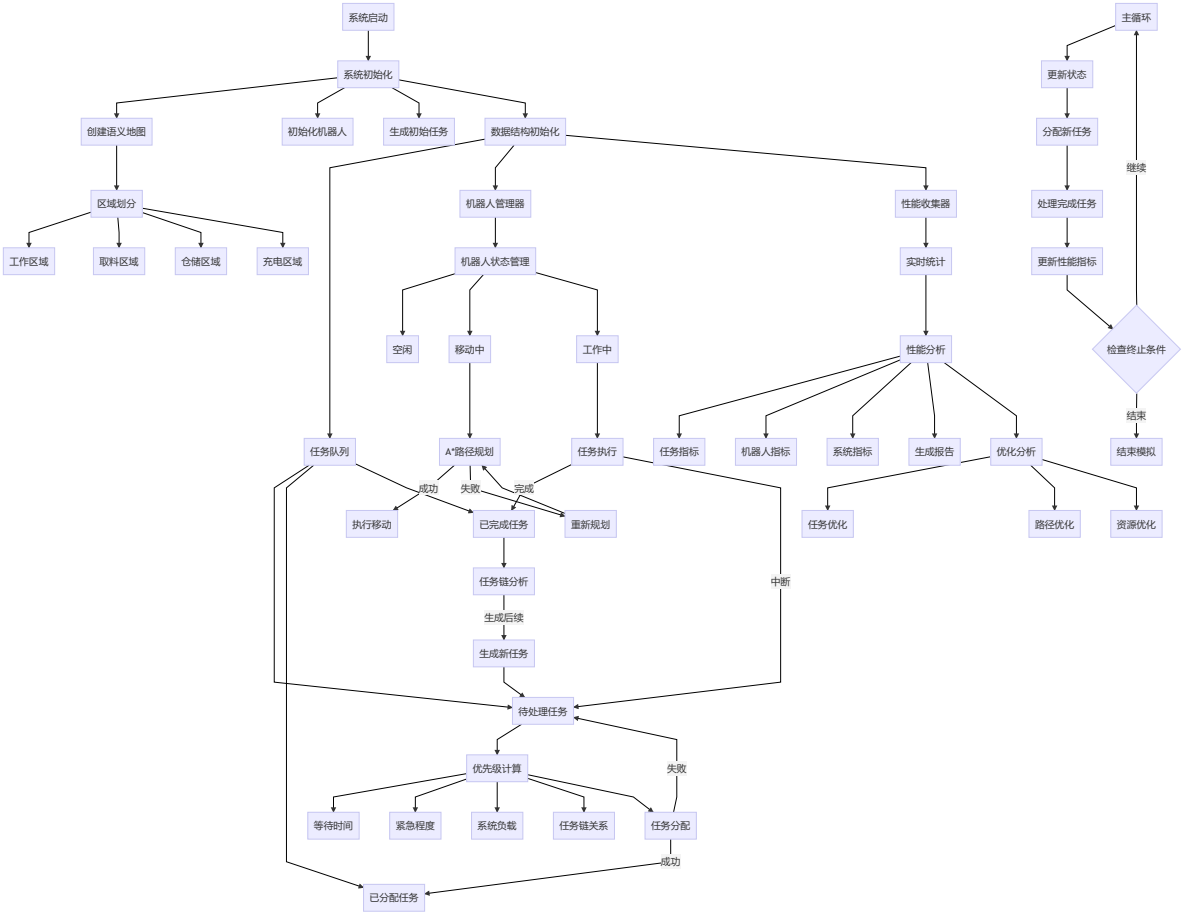
无锁


系统启动
系统初始化
创建语义地图
初始化机器人
生成初始任务
无锁数据结构初始化
机器人管理器
性能收集器
机器人状态管理
实时统计
空闲
移动中
工作中
性能分析
任务队列
碰撞检测
任务执行
生成报告
优化分析
无碰撞
有碰撞
完成
已完成任务
继续移动
避障
任务优化
路径优化
资源优化
路径规划
任务链分析
中断
成功
失败
需要生成
执行移动
重新规划
生成新任务
待处理任务
优先级计算
失败
任务分配
主循环
更新状态
分配新任务
错误检查
发现故障
否
处理完成任务
故障处理
更新性能指标
状态恢复
检查终止条件
负载均衡
是
结束模拟
优先级调整
机器人重分配

现在流程图更清晰了。您想从哪个组件开始实现？建议按以下顺序：

1. 核心数据结构（任务队列、机器人管理器等基础组件）
2. 任务管理系统（任务生成、分配、跟踪）
3. 机器人控制系统（状态管理、路径规划、避障）
4. 性能监控系统（数据收集、分析、报告）

有锁

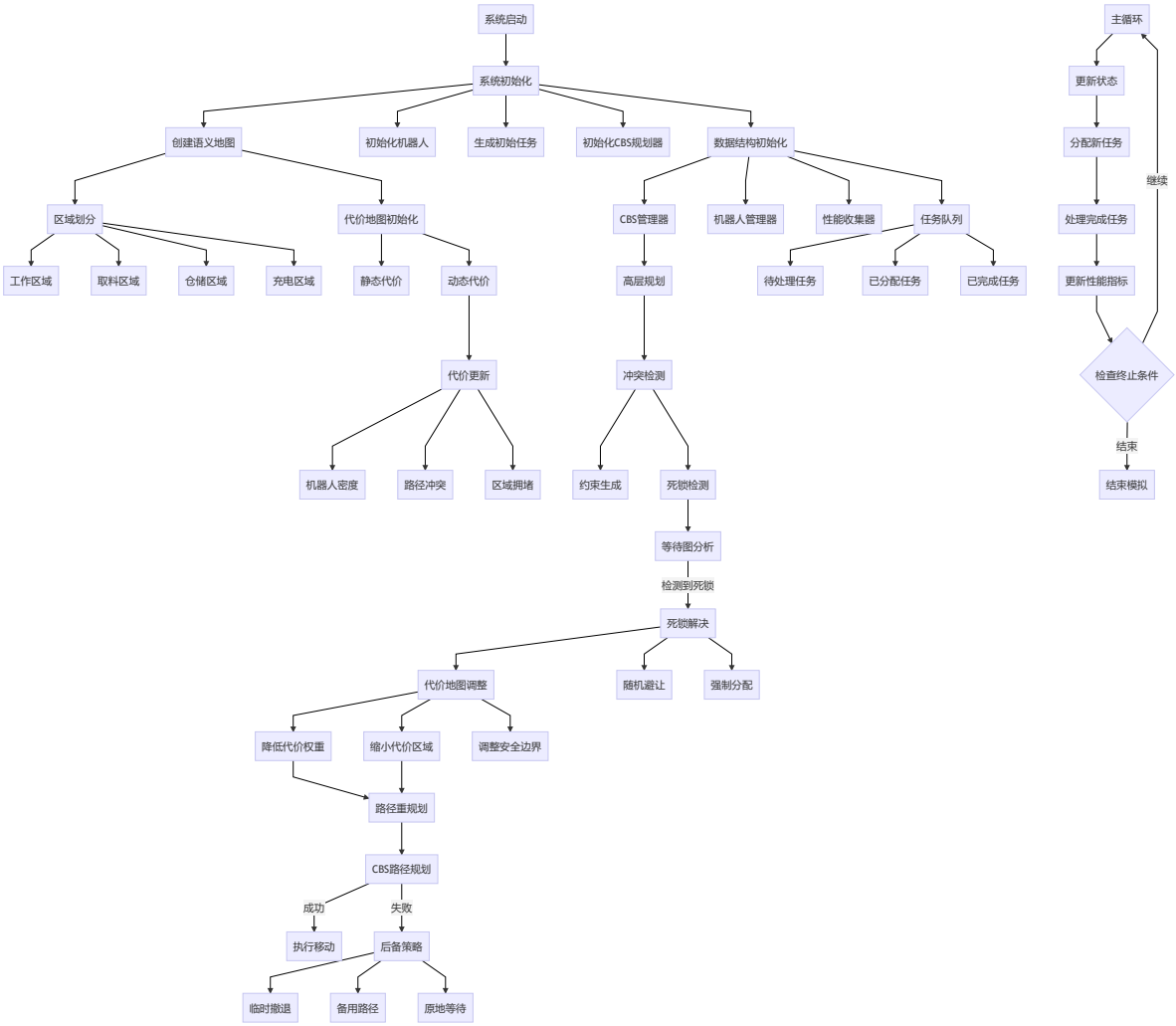

系统启动
系统初始化
创建语义地图
初始化机器人
生成初始任务
数据结构初始化
区域划分
工作区域
取料区域
仓储区域
充电区域
机器人管理器
机器人状态管理
空闲
移动中
工作中
性能收集器
实时统计
性能分析
任务队列
A*路径规划
任务执行
任务指标
机器人指标
系统指标
生成报告
优化分析
任务优化
路径优化
资源优化
成功
失败
完成
执行移动
已完成任务
重新规划
任务链分析
生成后续
生成新任务
中断
待处理任务
优先级计算
失败
等待时间
紧急程度
系统负载
任务链关系
任务分配
成功
已分配任务
主循环
更新状态
分配新任务
处理完成任务
更新性能指标
检查终止条件
继续
结束
结束模拟

这个完整的流程图包含了：

1. 主要流程：从启动到结束的完整流程
2. 初始化流程：系统启动时的设置过程
3. 模拟循环：主要的运行循环
4. 任务管理：任务的生命周期和优先级管理
5. 机器人控制：机器人的状态管理和路径规划
6. 性能监控：系统性能的监控和分析
7. 数据流：系统中的数据流动

CBS

我来重新设计一个完整的系统流程图，整合之前的优秀设计和新的CBS算法：


系统启动
系统初始化
创建语义地图
初始化机器人
生成初始任务
初始化CBS规划器
数据结构初始化
区域划分
代价地图初始化
CBS管理器
机器人管理器
性能收集器
任务队列
工作区域
取料区域
仓储区域
充电区域
静态代价
动态代价
高层规划
待处理任务
已分配任务
已完成任务
代价更新
冲突检测
机器人密度
路径冲突
区域拥堵
约束生成
死锁检测
等待图分析
检测到死锁
死锁解决
代价地图调整
随机避让
强制分配
降低代价权重
缩小代价区域
调整安全边界
路径重规划
CBS路径规划
成功
失败
执行移动
后备策略
临时撤退
备用路径
原地等待
主循环
更新状态
分配新任务
处理完成任务
更新性能指标
检查终止条件
继续
结束
结束模拟

这个系统流程图整合了：

1. CBS路径规划系统：

- 基于CBS的多机器人路径规划
- 冲突检测和解决机制
- 安全约束管理

1. 死锁处理策略：

- 通过随机避让解决死锁
- 设置临时避让区
- 强制分配机制

1. 任务管理优化：

- 任务链优化
- 动态优先级调整
- 多策略任务分配

1. 性能监控和优化：

- 实时性能监控
- 自适应策略调整
- 资源负载均衡

1. 异常处理机制：

- 状态备份和恢复
- 紧急处理流程
- 系统容错

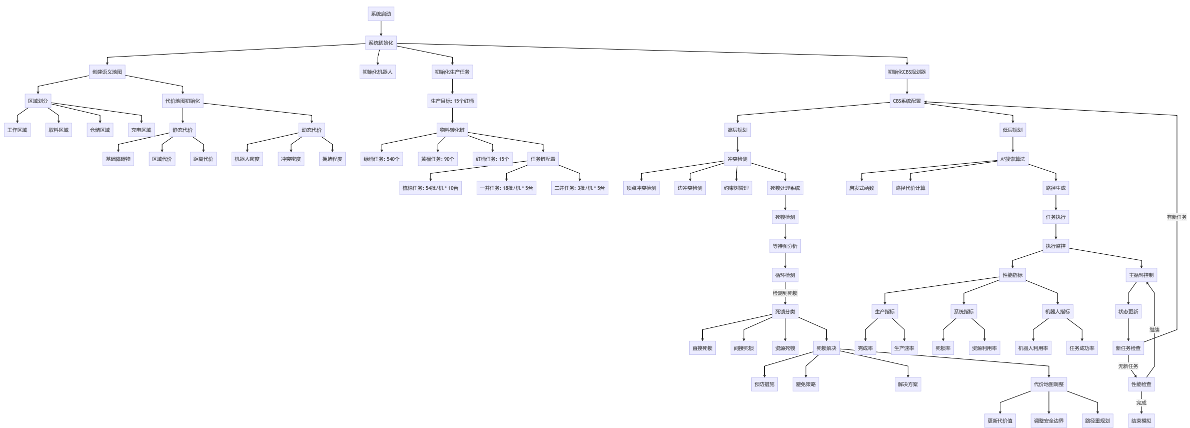
需要我详细说明某个具体模块的实现吗？我可以为任何模块提供详细的代码实现。

另外，对于死锁问题的具体解决方案，我建议结合CBS和随机避让策略：

1. 首先使用CBS进行全局路径规划
2. 当检测到死锁时，使用随机避让策略
3. 如果随机避让失败，则启用强制分配机制

这样可以在保证效率的同时提高系统的鲁棒性。需要我详细展开这部分的实现吗？

您说得对，我重新整合之前的CBS和死锁部分，生成一个更完整的系统流程图：



系统启动
系统初始化
创建语义地图
初始化机器人
初始化生产任务
初始化CBS规划器
区域划分
代价地图初始化
生产目标: 15个红桶
CBS系统配置
工作区域
取料区域
仓储区域
充电区域
静态代价
动态代价
物料转化链
高层规划
低层规划
基础障碍物
区域代价
距离代价
机器人密度
冲突密度
拥堵程度
绿桶任务: 540个
黄桶任务: 90个
红桶任务: 15个
任务链配置
冲突检测
A*搜索算法
梳棉任务: 54批/机 * 10台
一并任务: 18批/机 * 5台
二并任务: 3批/机 * 5台
顶点冲突检测
边冲突检测
约束树管理
死锁处理系统
启发式函数
路径代价计算
路径生成
死锁检测
任务执行
等待图分析
执行监控
有新任务
循环检测
性能指标
主循环控制
检测到死锁
死锁分类
生产指标
系统指标
机器人指标
状态更新
继续
直接死锁
间接死锁
资源死锁
死锁解决
完成率
生产速率
死锁率
资源利用率
机器人利用率
任务成功率
新任务检查
无新任务
预防措施
避免策略
解决方案
代价地图调整
性能检查
完成
更新代价值
调整安全边界
路径重规划
结束模拟

这个完整流程图包含了：

1. CBS路径规划完整流程
    - 高层规划：冲突检测、约束树管理
    - 低层规划：A*搜索、启发式函数
    - 代价计算和路径构建
2. 死锁处理系统
    - 检测机制：等待图、循环检测
    - 死锁类型：直接、间接、资源死锁
    - 解决策略：预防、避免、解决
3. 生产任务链
    - 物料转化：540绿桶 -> 90黄桶 -> 15红桶
    - 处理时间：梳棉4225.94s、并条2594.75s
    - 任务批次：梳棉54批、一并18批、二并3批
4. 完整的监控系统
    - 生产指标
    - 系统指标
    - 机器人指标

需要我开始代码实现吗？我会基于这个详细流程图，确保每个组件都得到完整实现。