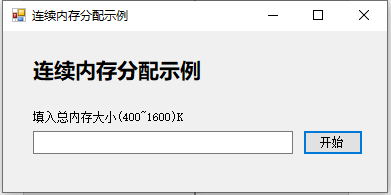
# 内存管理设计方案报告

## 项目要求

初始态下存在一请求序列，请分别用首次适应算法和最佳适应算法进行内存块的分配和回收，并显示出每次分配和回收后的空闲分区链的情况。

## 使用说明

1、运行程序，输入要模拟的内存大小，进入主界面



2、 程序左上角为编辑进程调度队列模块，可采取随机生成调度队列或自定义调度队列的方式。生成随机队列会清空以前队列里的内容，可在生成随机队列后再添加要创建调入或释放的进程。

可点击主界面右侧清空进程队列按钮清空调度队列。



3、 调度算法可选首次适应算法或最佳适应算法。点击左侧执行单步或执行全部按钮开始执行。



4、 点击重置按钮，程序恢复至初始状态

## 本程序运行规则

1. 调度队列长度最大为30。
2. 可创建的进程大小范围为：总内存容量的2.5% 至300K。
3. 生成随机队列时先让内存充满至大于70%，再进行随机操作
4. 随机状态下创建新调入进程的概率为（剩余内存空间 / 总内存空间 \* 1.2）
5. 执行调入或释放操作后（并非指添加进调度队列）的进程在重置系统之前不会再被重新创建。
6. 单步执行时可往进程队列添加或释放进程
7. 执行完整个调度队列后，调度队列才会被清空。
8. 内存占用表与空闲表的数据结构为双向链表
9. 为更好地呈现空闲表与占用表的信息，两表内容均按内存地址顺序排列。

## 代码实现

本程序用C#语言通过winform框架来实现。控件操作均来自winform框架。

主要源代码在MainForm.cs，MainForm.Designer.cs，StartForm.cs，StartForm.Designer.cs中。

### 类

**StartForm类：**程序开始界面，主要包含内存大小输入框与开始按钮，获取输入的内存大小信息并创建程序主界面

**MainForm类：**程序主界面，包含所有的内存调度的UI及交互，运行逻辑等，下方将详细介绍

**Process类：**进程类，存储创建的进程的信息，如编号，大小，在内存中的位置及其UI模块

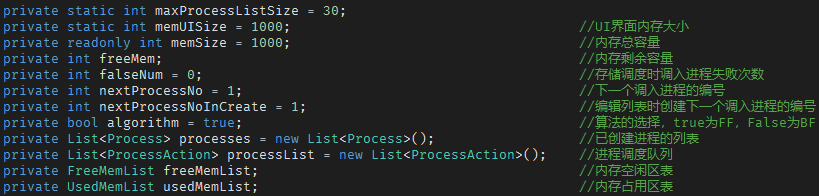
**ProcessAction类：**进程操作类，存储进程的行为（调入/释放）及对应的进程

**FreeMemList类：**内存空闲区表，存储内存空闲区表的表头

**UsedMemList类：**内存占用区表，存储内存占用区的表的表头

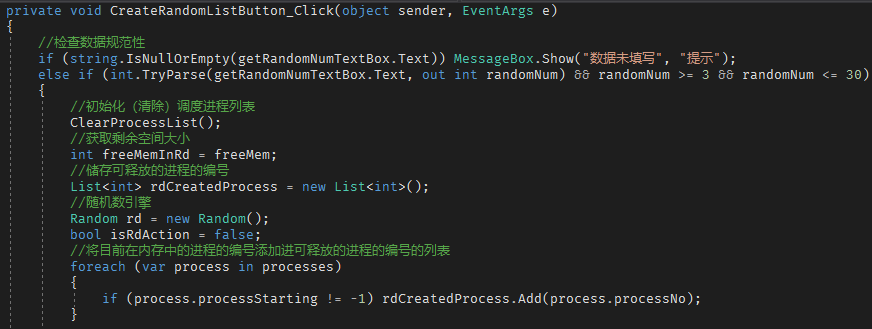
**MemBlock类：**内存块表，作为空闲区表占用区表两个双向链表的结点，存储在内存中的位置及其大小与他的上一个或下一个内存块的信息

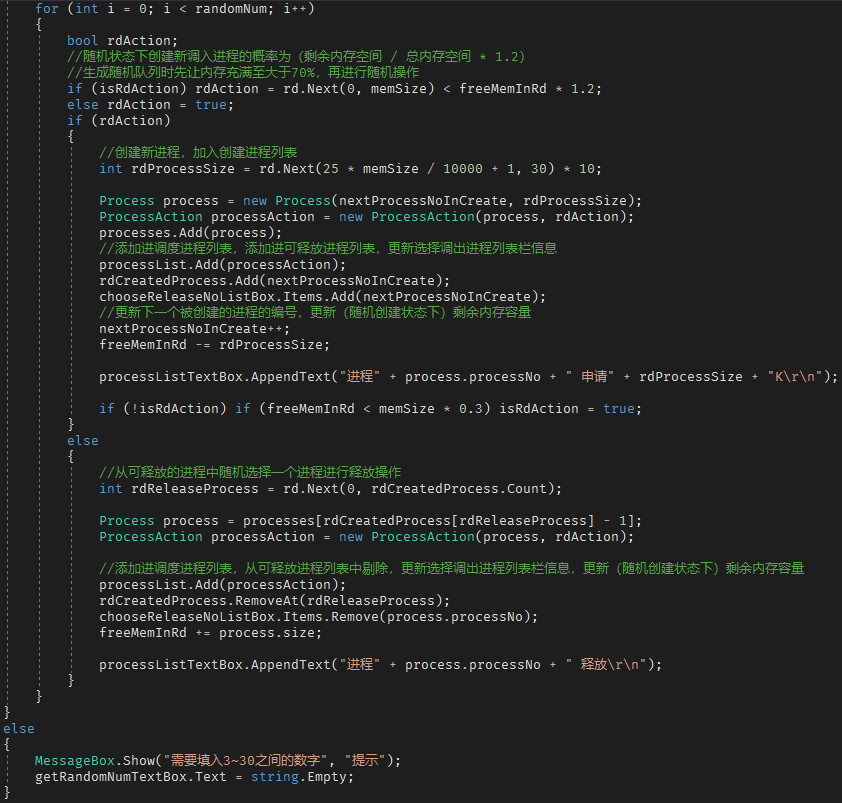
### MainForm类主要属性



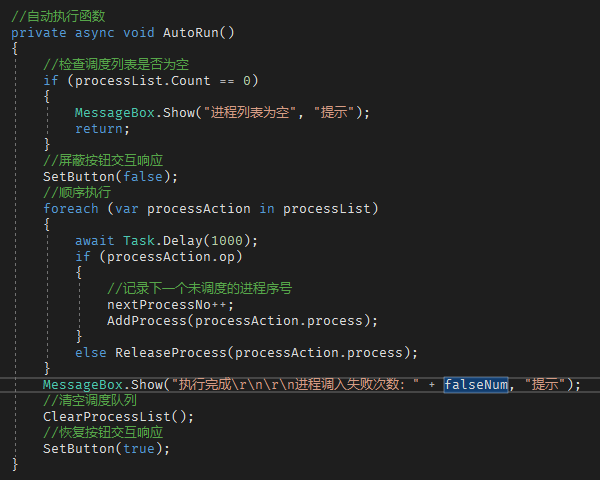
### 主要运行逻辑

#### 生成随机调度列表

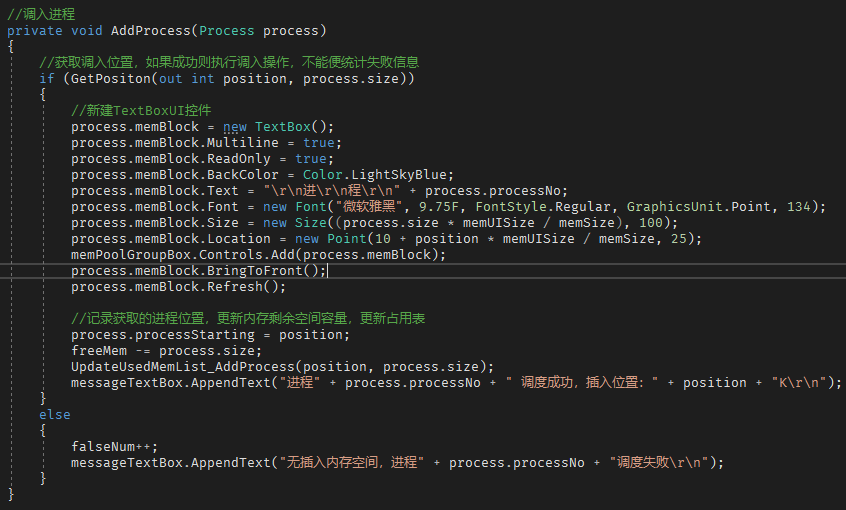




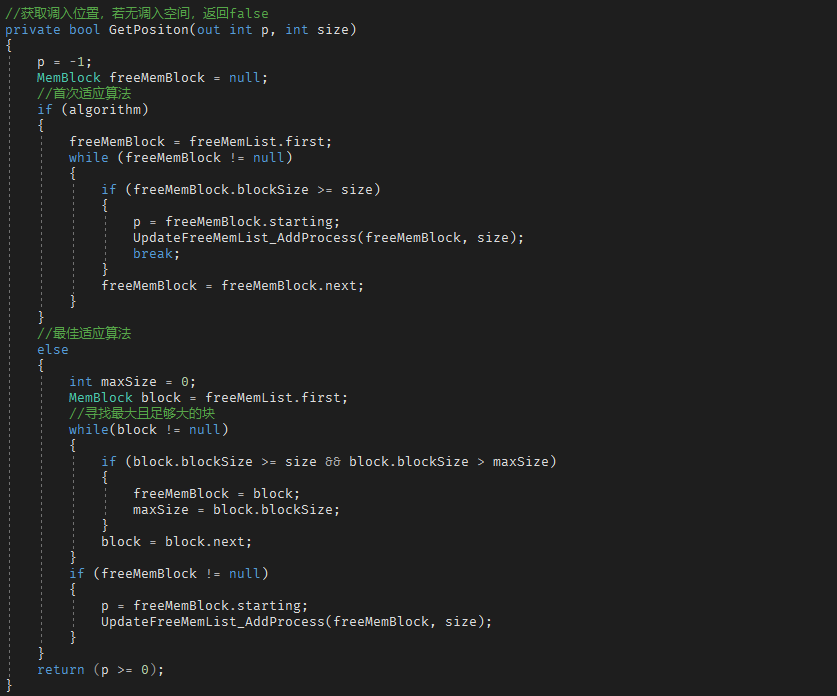
#### 自动执行函数



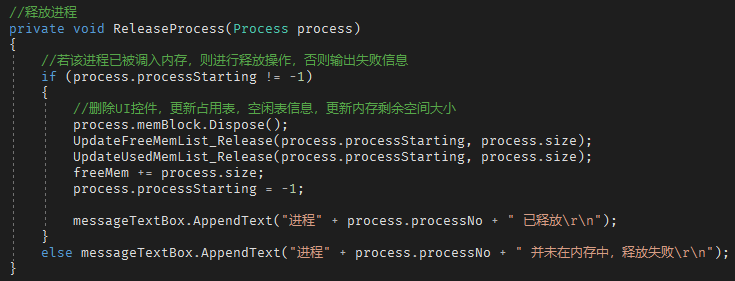
#### 调入进程操作



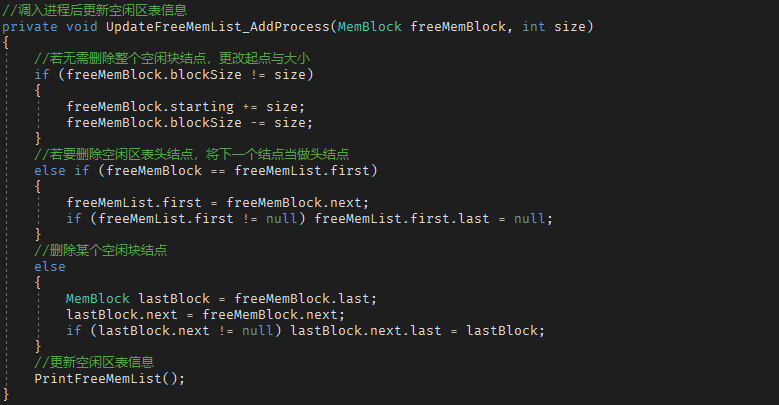
#### 获取调入进程的位置

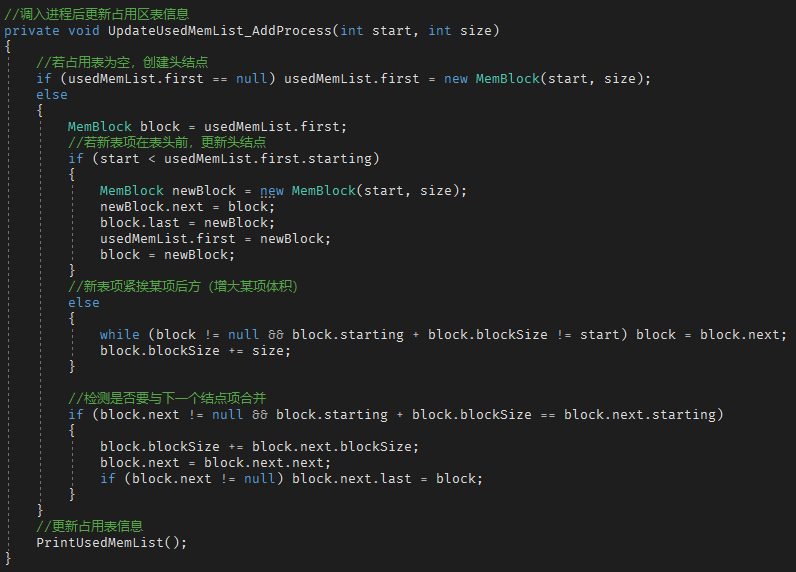


#### 释放进程操作

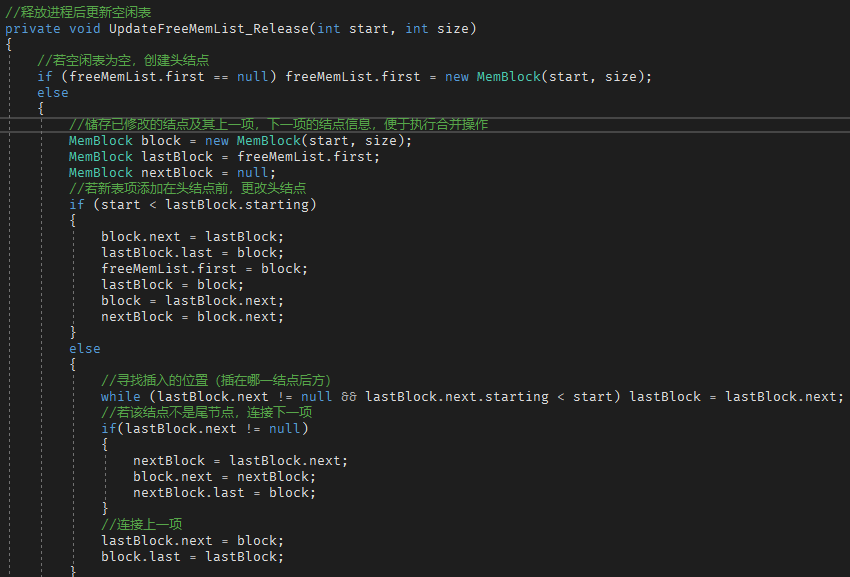
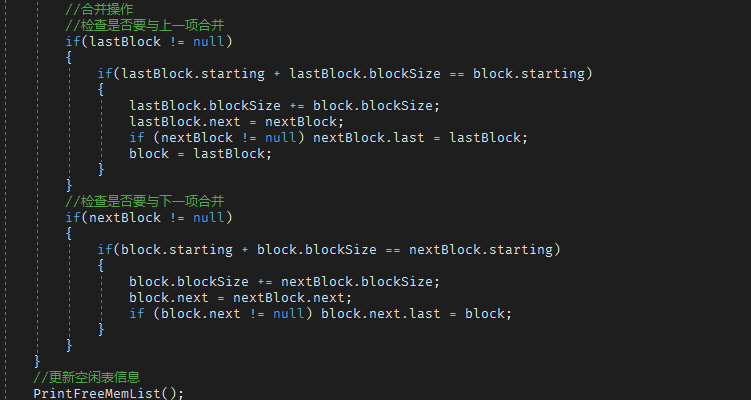
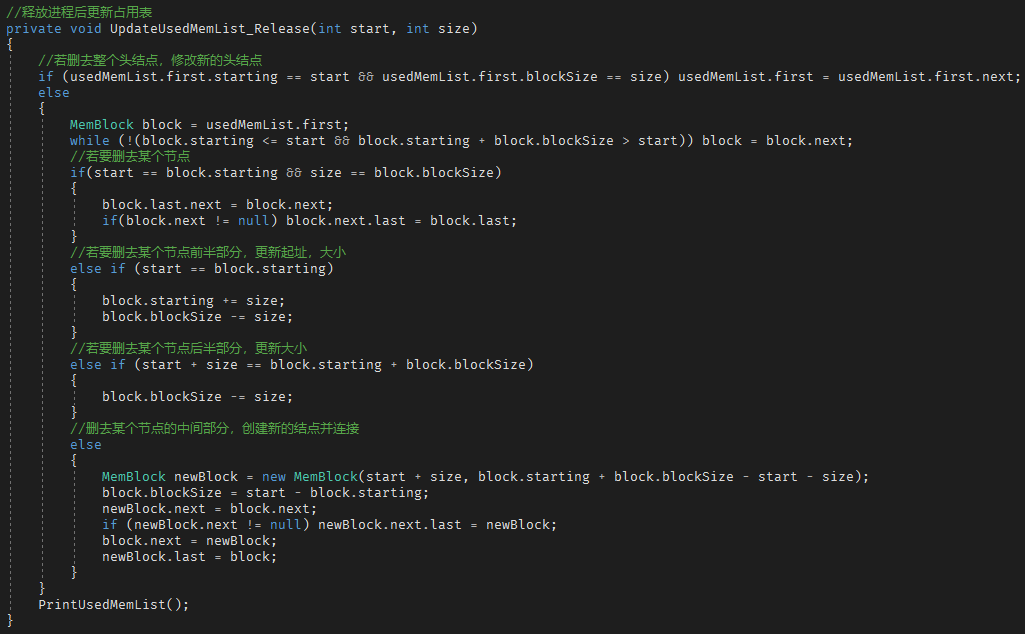


#### 调入操作后，更新两表信息

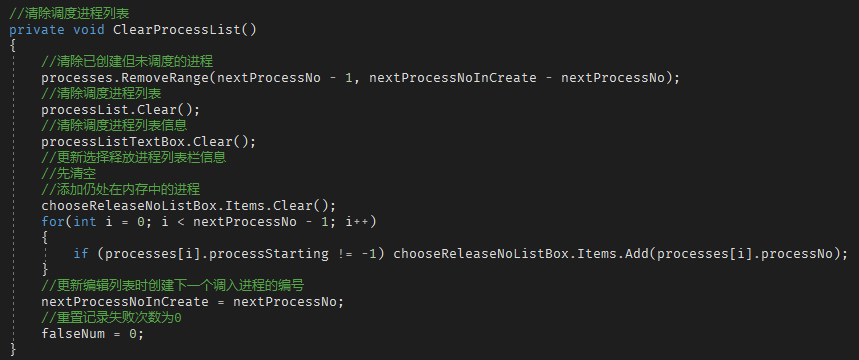




#### 释放操作后，更新两表信息

#### 清除调度进程列表操作



#### 重置系统操作

