# 内存管理项目

2150271 吴可非

## 项目简介

本项目实现了一个简单的内存管理模拟器，使用 构建了一个可视化界面。用户可以输入进程块大小并选择内存分配算法（首次适应或最佳适应），然后观察内存布局和内存块链表的变化。

## 界面展示



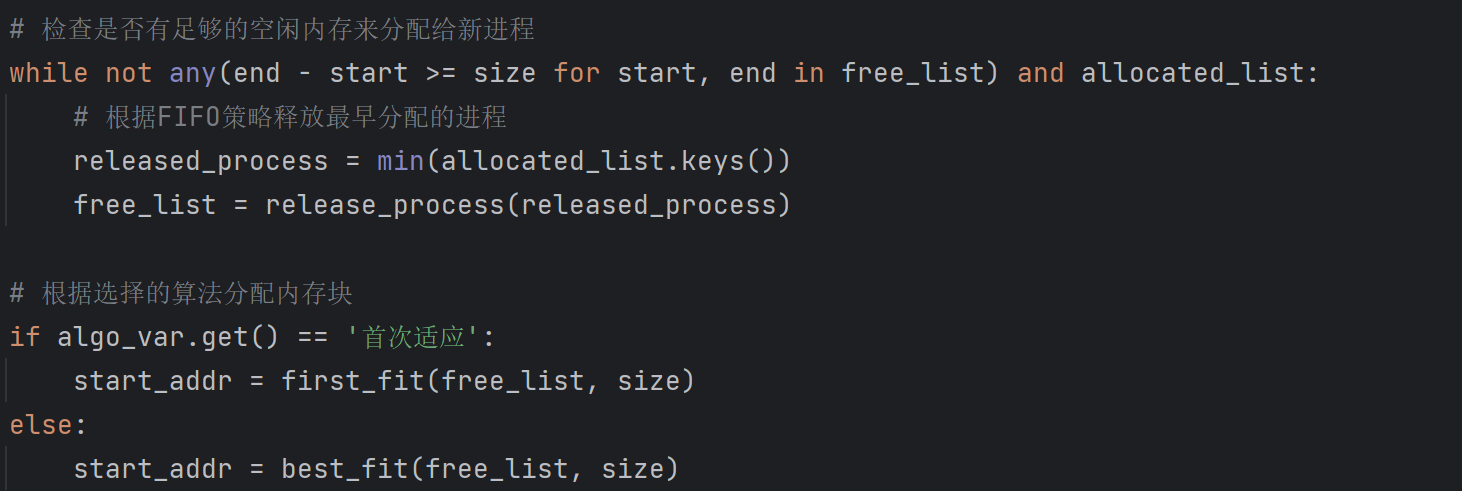
## 项目实现

1. 导入所需库，包括 和 。

2. 初始化一些全局变量，比如进程计数、已分配内存列表、 根窗口、内存布局画布、算法选择变量、进程块大小输入框、空闲内存列表和内存块链表表格。

3. 函数定义了应用程序的整个图形用户界面，包括算法选择按钮组、进程块大小输入框、提交和清空按钮、内存布局画布、内存块链表表格和退出按钮。该函数最后调用 来启动 事件循环，使应用程序运行并响应用户输入。

4. 函数分配内存给新进程。它首先获取用户在输入框中输入的进程块大小，然后检查是否有足够的空闲内存。如果没有足够的空闲内存，它将根据策略释放最早分配的进程，以便为新进程释放空间。然后，根据用户选择的算法（首次适应或最佳适应），分配内存，并更新内存布局和内存块链表。



5. 函数负责释放给定进程号的内存，并更新空闲内存列表。它还负责合并相邻的空闲内存块。

6. 和 函数分别实现了首次适应和最佳适应算法。它们根据输入的空闲内存列表和所需的内存大小，返回合适的内存块起始地址。

7. 函数根据给定的起始地址和大小，将内存分配给新进程，并更新空闲内存列表。

8. 和 函数负责在画布上显示内存布局以及更新并显示内存块链表。

9. 函数是一个回调函数，当用户点击清空按钮时调用。它会清空内存布局画布、重置内存块链表为空、重置已分配内存列表并清空表格。

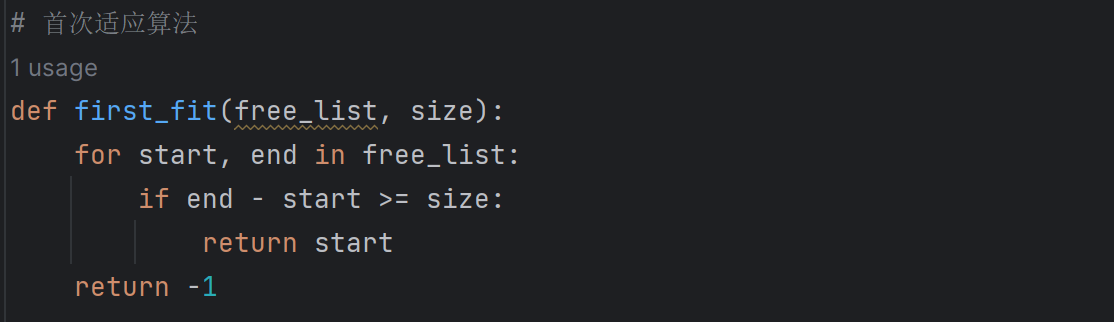
10. 最后调用 函数来启动应用程序。

## 算法

在内存管理模拟器中，实现了两种内存分配算法：首次适应（First Fit）和最佳适应（Best Fit）。这两种算法都是用于在动态分区存储器中分配内存的策略。

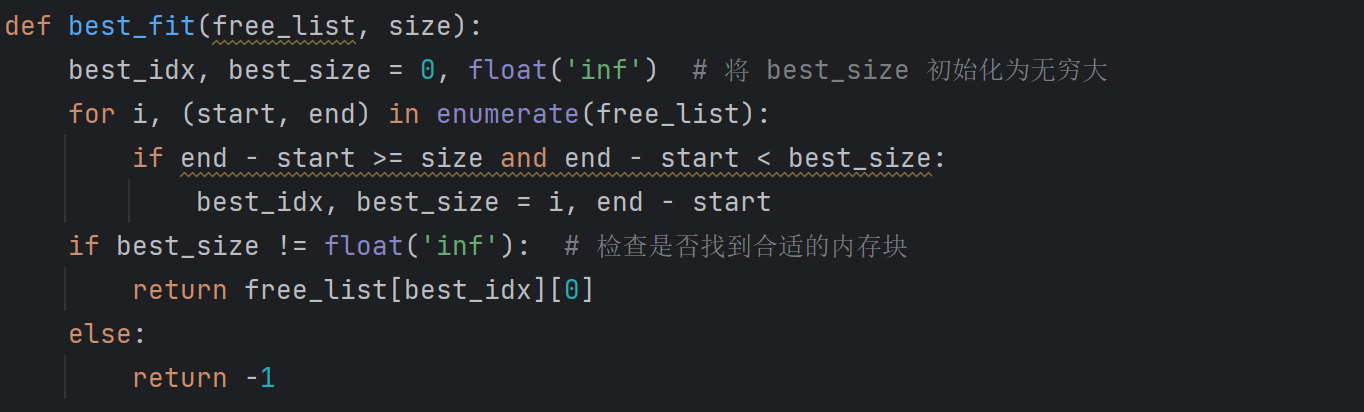
1. 首次适应算法：

函数接收两个参数：一个空闲内存列表和一个所需内存大小. 它遍历空闲内存列表，找到第一个空间足够大的内存块。一旦找到合适的内存块，函数返回该内存块的起始地址。如果没有找到合适的内存块，函数返回-1。



1. 最佳适应算法：

函数同样接收两个参数：一个空闲内存列表和一个所需内存大小 它遍历空闲内存列表，找到一个足够大且与所需内存大小最接近的内存块。在遍历过程中，它使用一个变量 来跟踪当前找到的最佳内存块与所需内存大小的差值。一旦遍历完空闲内存列表，函数返回找到的最佳内存块的起始地址。如果没有找到合适的内存块，函数返回 -1。



## 操作指南

1. 选择分配算法:首次适应或最佳适应。选择对应的即可。
2. 输入进程块大小并提交:在输入框中输入进程块大小,点击“提交”按钮进行内存分配。



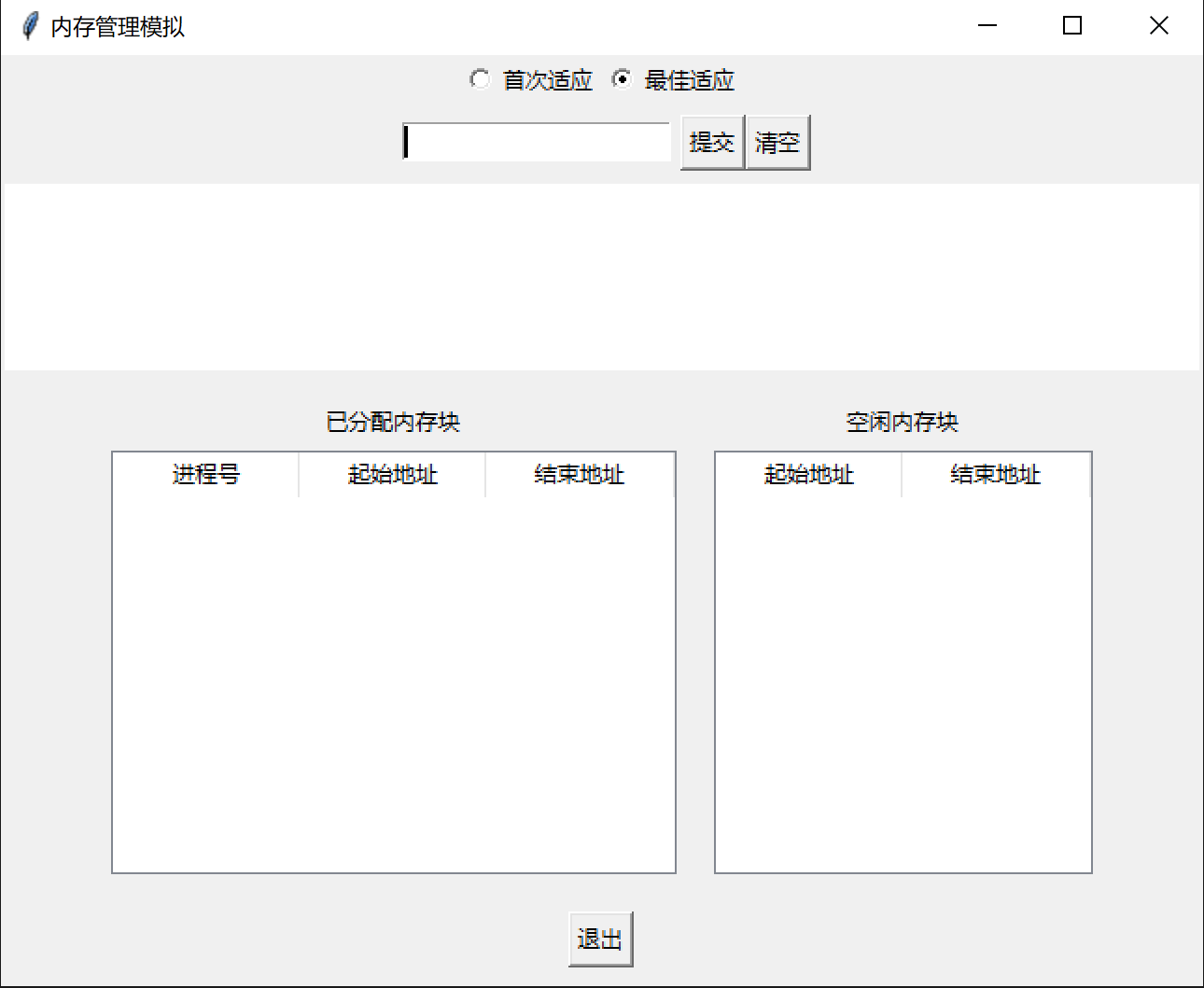
1. 组件用来显示当前内存布局,已分配的内存块为蓝色矩形,空闲内存块为灰色矩形。



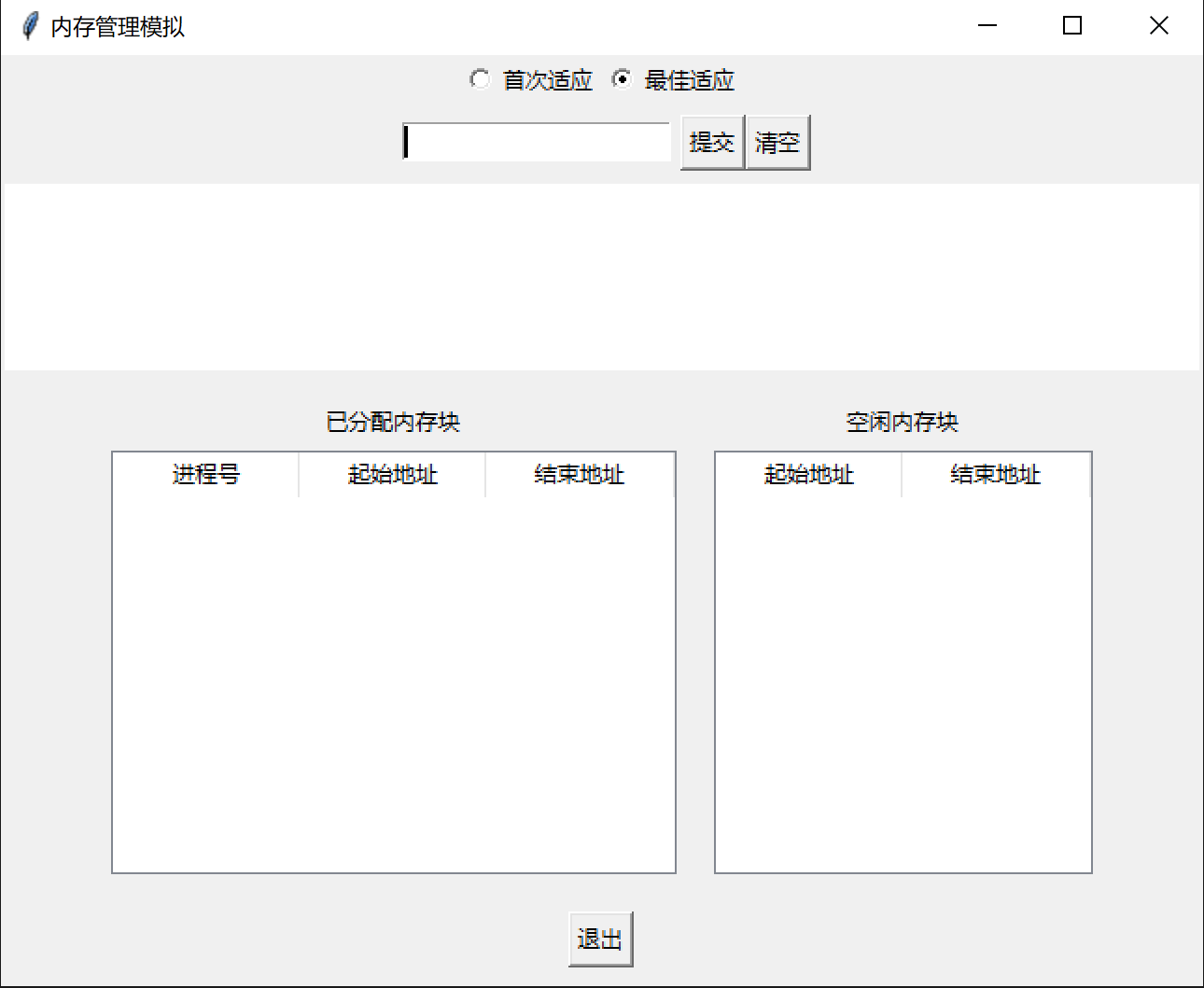
1. 组件显示当前空闲内存块链表,显示起始地址和结束地址。



1. 当内存不足以分配新进程时,会根据策略释放最早分配的进程,以获取更多空闲内存。
2. 点击“清空”按钮可以清空当前内存布局和内存块链表,重置内存管理器。



1. 退出:点击“退出”按钮退出内存管理器程序。



可以观察到，对于两块空闲空间，申请内存比第二块空闲空间小时：

如果用户选择首次适应，分配到先找的大空间；

如果选择最佳适应，则会选择较小的第二个空间。



