动态分区分配

项目目标：

项目旨在模拟内存管理中的动态内存分配机制，以更好地理解动态内存分配中的不同算法的使用的利弊。

项目需求：

1. 根据用户在进行模拟前需求，动态地选择模拟的内存大小，如果未输入选择模拟分配的内存大小，则默认模拟内存大小为640K。
2. 用户可以导入内存分配的指令的文本，当然也可以选择默认的指令来模拟过程。
3. 模拟过程中实时地显示出内存分配的状态，与指令分配的情况。
4. 对于模拟过程中，内存分配中未能及时分配的内存块，单独显示出来。

项目实现：

数据结构：

实现过程中维护了一个作业内存信息的表和空白内存表，每块内存的信息由MemBlock类维护，MemBlock中分别维护了内存块的开始位和结束位，同时记录了改内存块为空白内存还是已被使用的内存。

算法实现：

项目提供了首次适应算法和最佳适应算法两种算法用于动态内存分配。

首次适应算法

首次适应算法将从头开始遍历空闲分区表，把最先能够满足要求的空闲区分配给作业，这种方法目的在于减少查找时间。

最佳适应算法

最佳适应算法是从空闲区中找出能满足作业要求的、且大小最小的空闲分区的一种计算方法，这种方法能使碎片尽量小。算法将搜索所有空闲的分区块，寻找最合适的一块将其分配。

实现细节：

空闲内存分区

将一个空闲内存分配给作业，将把给空闲内存中从开始的内存地址开始到满足作业要求的内存大小的地址分配给作业，剩余的内存重新加入空闲内存。

作业释放内存

作业释放内存后，将会从作业被分配的内存区间前后，搜索是否存在空闲内存，如果存在则将其释放后内存和空闲内存合并，重新维护内存表。

未分配的内存处理

对于由于没有合适的空闲内存而暂时无法分配空间的内存块，将维护一个保存未分配的内存的作业表。每次有作业释放内存时，将优先查看是否有作业表中是否有可以分配的作业。

文件说明

环境说明

本项目使用java1.8实现，实现后使用Intellij Idea来编译java，并将其到处为jar文件。

经试验，jar文件在windows、linux均可顺利运行。

代码说明

MemSystem