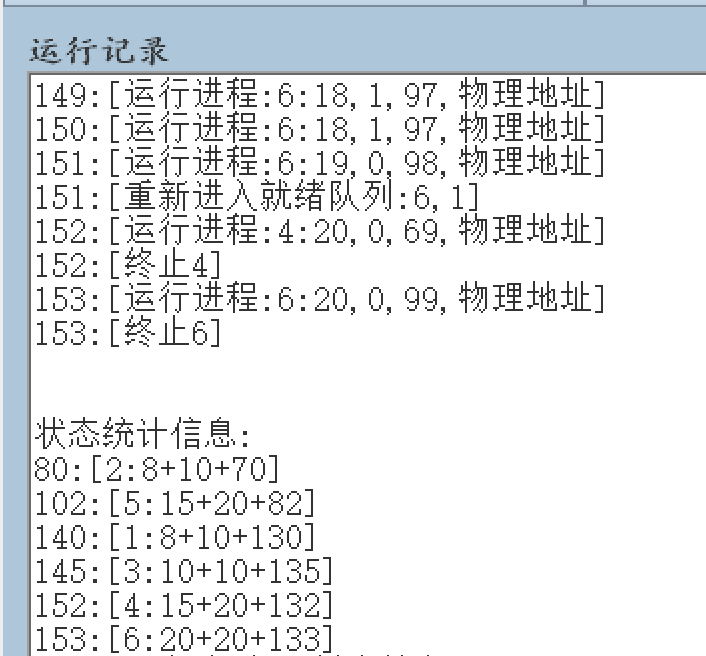
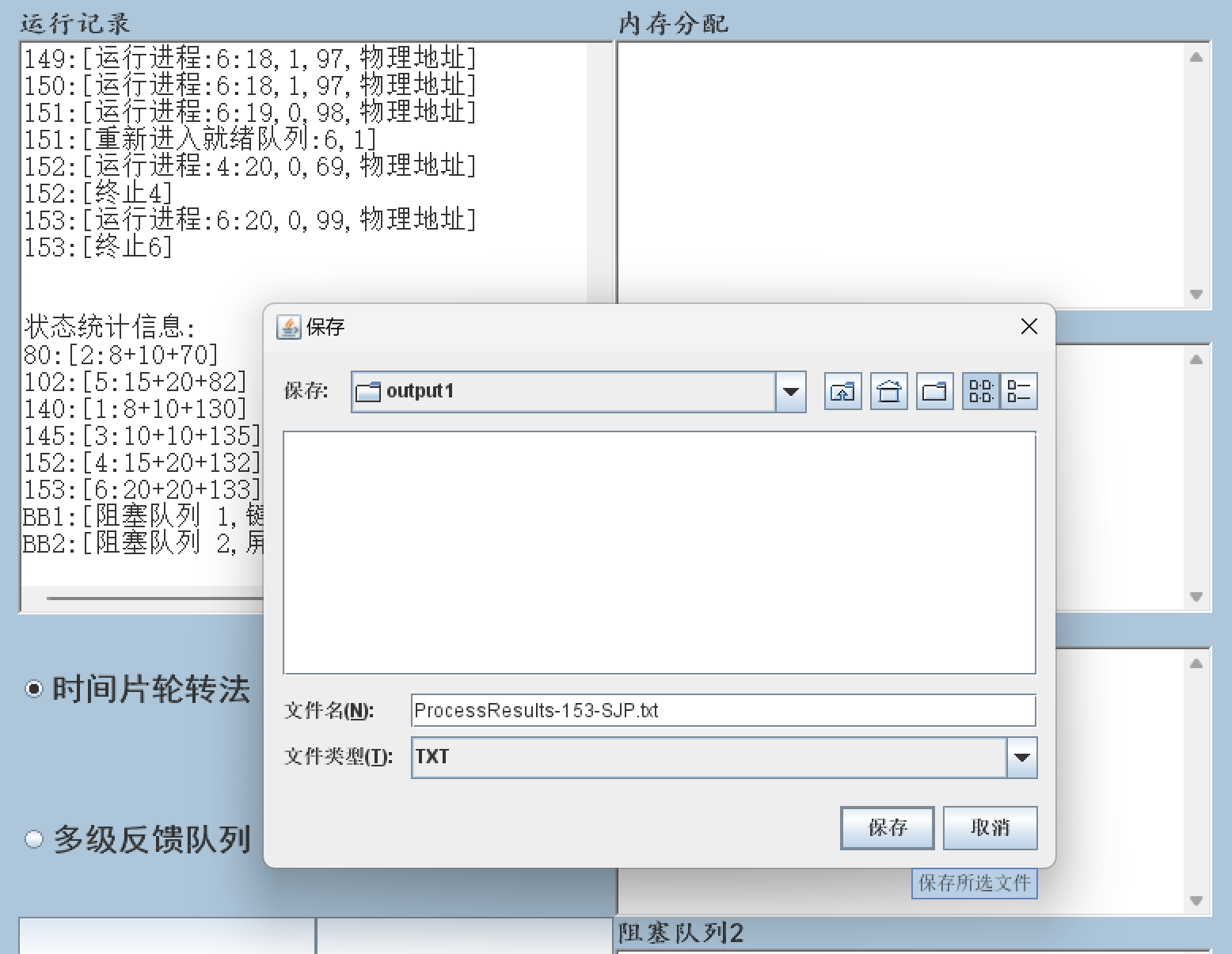
**10.10课设总结**

1. 完成任务

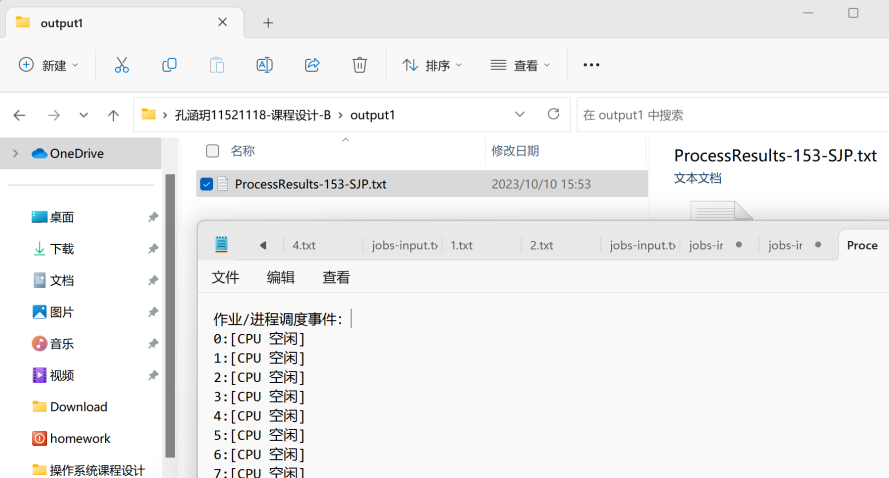
1.对于时间片轮转法，通过input1文件的测试。



2.对于文件导出的格式：



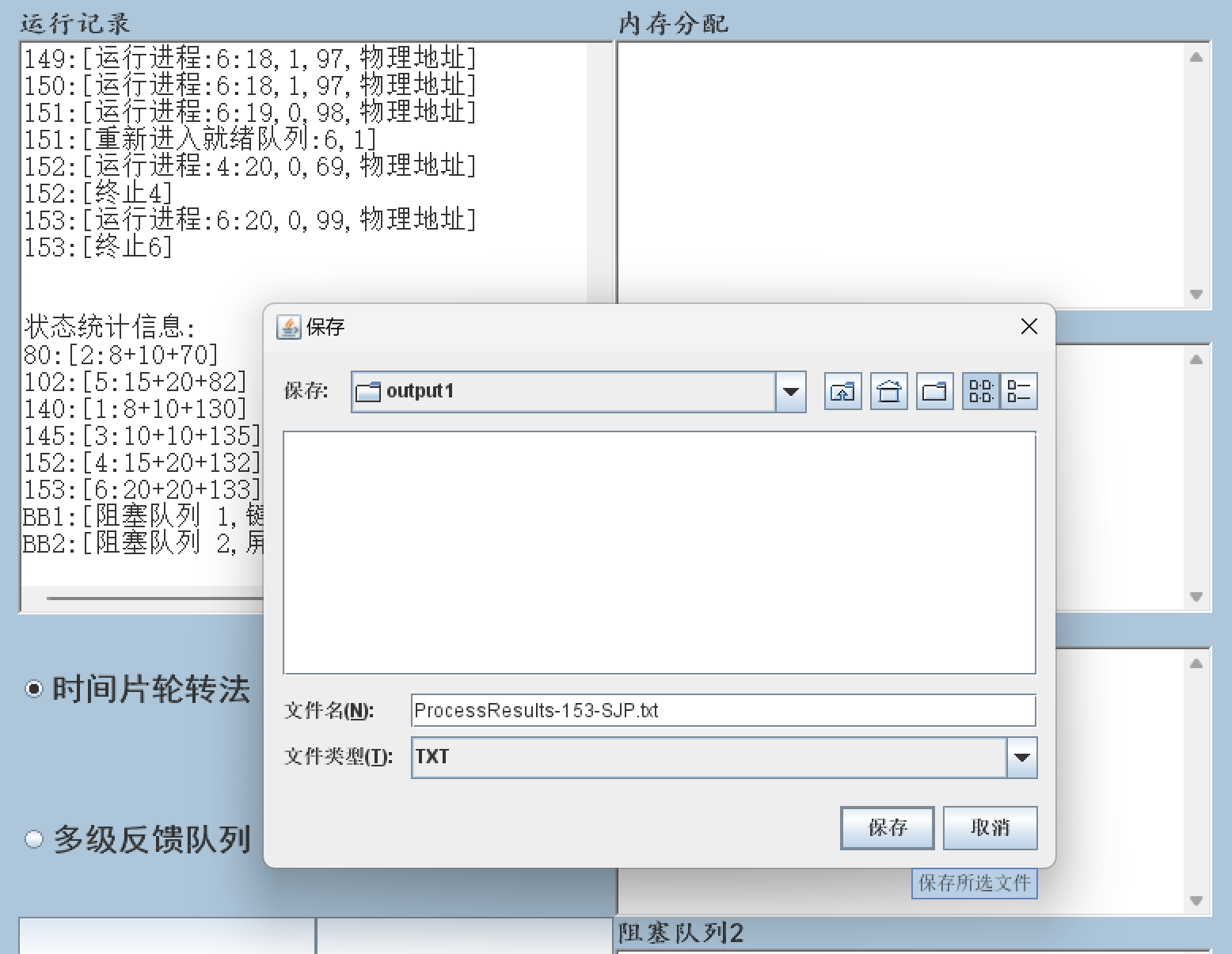
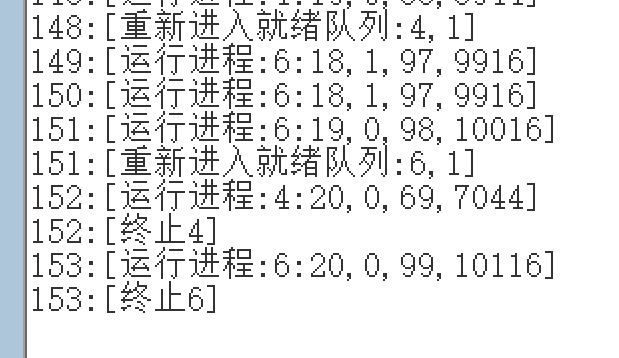
运行完成，点击导出后，可以在对应文件夹中找到：



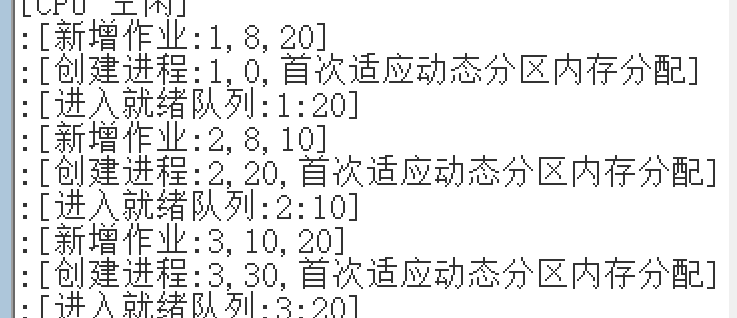
3.对于指令物理地址的显示部分：

指令的逻辑地址到物理地址的转换已经完成，指令的逻辑地址=指令所属的PCB所在的逻辑存储页面的页号\*10+指令在进程中的序号，通过指令所在逻辑页面的页号，查询页表找到指令所在的物理页面的页框号，乘以页面大小1024B，再与一条指令的长度100B\*指令在本页的序号相加，所得即是指令的物理地址：

修改前： 修改后：

1. 对于JCB转换为PCB时，涉及的信息输出：“1:[创建进程:进程 ID,PCB内存块始地址,内存分配方式]”中，“内存分配方式”部分，在查找资料：“内存分配方式及分配算法优劣”后确定我采用了首次适应动态分区内存分配的内存分配方法，将信息输出修改为：



1. 遇到的问题与解决方式：

1.对于进程运行情况的显示：

当程序运行时，若一个进程正在占用CPU（本不应该在就绪队列里），但会出现在就绪队列情况显示框中。

比如：

进行排错之后，确定了并非PCB未完全从就绪队列出队的问题，然后发现问题出现在就绪队列显示框的显示函数：

    public static boolean Check\_empty\_Ready() // 就绪队列现状

当就绪队列为空时，满足if(List\_Ready.size() == 0)的条件，因此直接进入if分支，直接执行指令 return false;了，因此无法对显示框起到清空作用。

解决方式：

在if语句分支中补充指令ui.textArea\_ready.setText("");先清空文本框再return 0;结束此函数。修改后可以真正“实时”正确地显示各队列的实时运行情况。

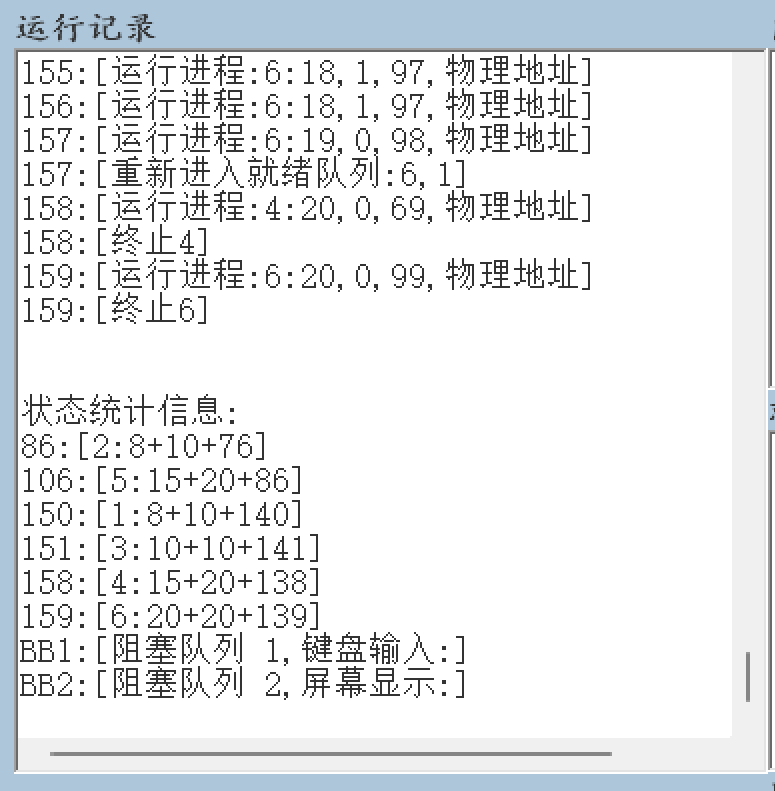
2.在显示文件与进程调度信息的时候，当一个作业转变为进程，刚进入就绪队列时会出现两条输出信息：“2:[进入就绪队列:进程 ID:待执行的指令数]”与“8:[重新进入就绪队列:进程 ID,待执行的指令数]”。

经过检查和排错后，发现并不是因为进程一次性进入了两次就绪队列，而是因为进入就绪队列时的输出语句未条件判断。

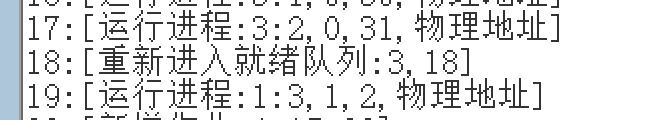
修改方式：

对于“8:[重新进入就绪队列:进程 ID,待执行的指令数]”的输出，先进行条件判断：if(PCB\_manage.PCB\_table.get(index).PC != 0)，当程序计数器的值不为０（已经执行过指令）后，才可以输出“8:[重新进入就绪队列:进程 ID,待执行的指令数]”信息。

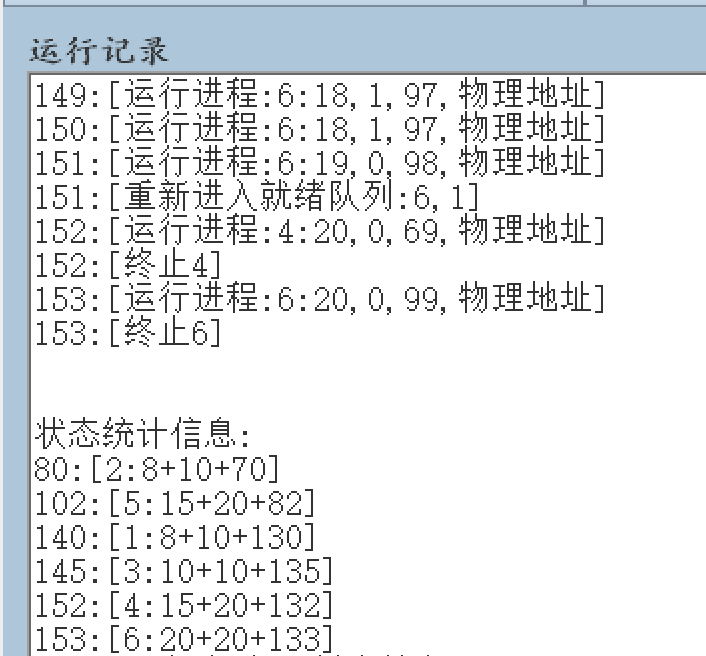
3.在使用input1文件夹测试时间片轮转法时，出现了运行总时长超出的问题。



经过逐项核对以后发现并非进程或作业调度、函数调用顺序的问题，而是“重新进入就绪队列执行函数”占用了1s的时间片，比如：



因此将占用的时间片回溯后即可解决问题：



1. 待解决的问题：
2. 进程调度部分，完成多级反馈队列的进程调度方法并通过input2的测试
3. 文件打开部分，将本系统中所有的设计文件地址的部分与“打开”按钮点击后所选择的文件和文件夹相匹配，实现手动选择输入文件。
4. 文件打包为.exe的尝试。