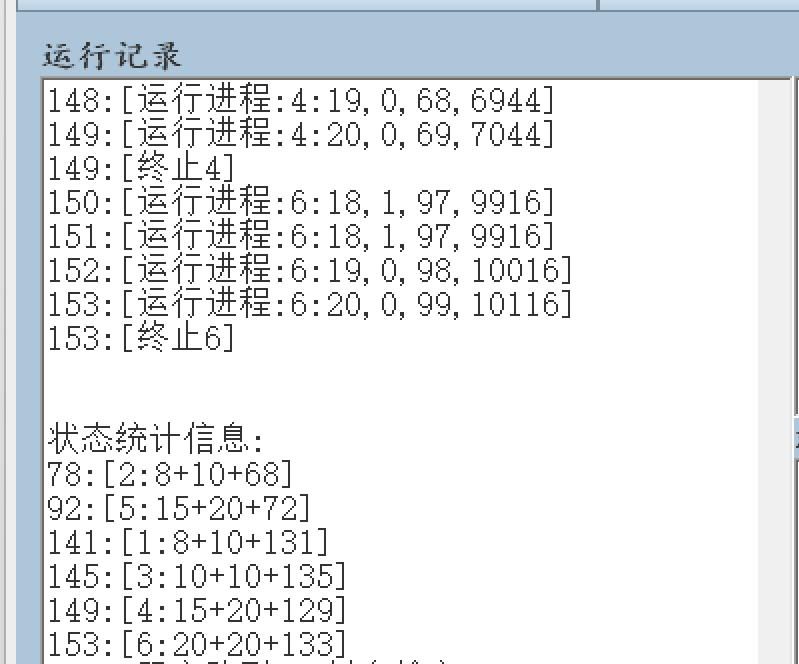
**10.11课设总结**

1. 完成任务
2. 对于多级反馈队列算法，通过input1文件的测试：

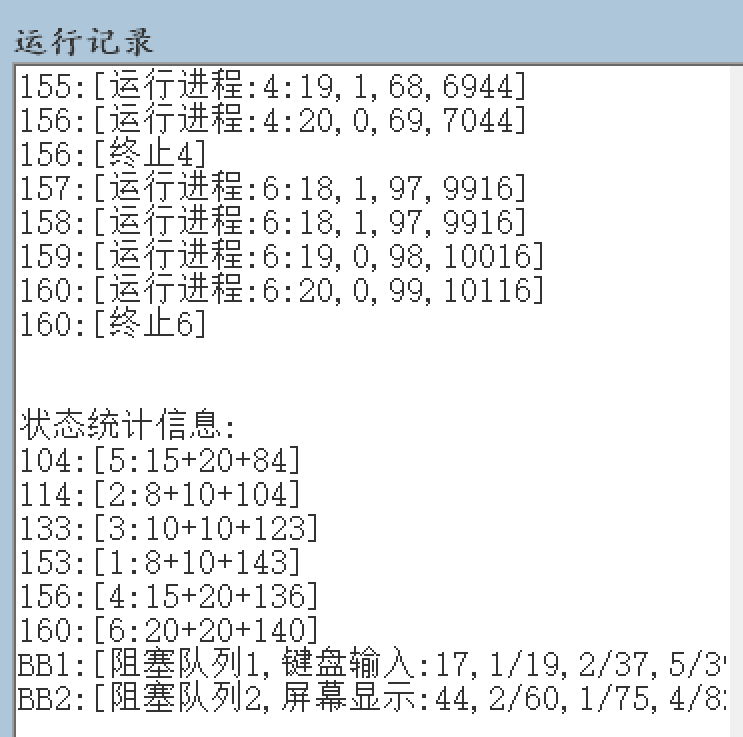
设计了三个“队列”，在代码中以“优先级”的方式呈现，数字分别为5，4，3，对应的时间片大小依次为3s，4s，5s。



1. 对于多级反馈队列算法，通过input2文件的测试：

设计了两个线程：InputBlock\_thread与OutputBlock\_thread，用于阻塞队列（IO）的管理，并将这两个进程加入原有的Clock\_thread、JobIn\_thread、ProcessScheduling\_thread三线程并发的模型中，达成5线程并发。

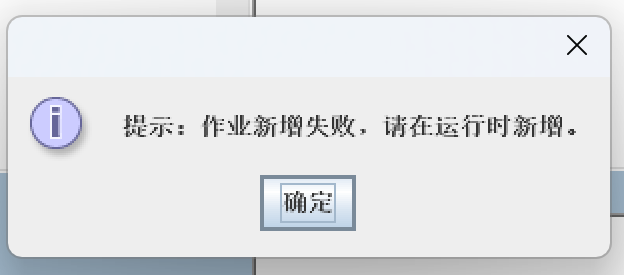
对于IO的两个线程，设计了阻塞队列出队判断函数CHECK\_INPUT()与CHECK\_OUTPUT()用于判断当前时间是否有可以出队的节点，并进行出队。



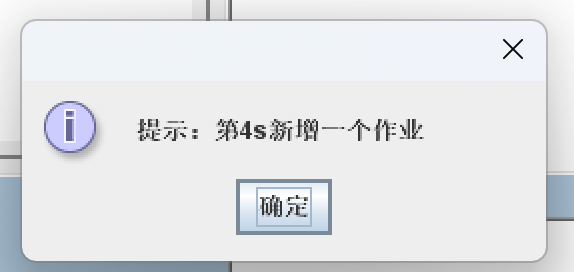
1. 对于实时点击按钮新增作业：

设计随机函数，对于实时的点击，随机设置此作业中的指令条数insnum，范围为（10-20），随机设置此作业中的指令种类，范围为（0-3），顺序分配一个空的作业编号，作业请求时间即为按钮点击时间。

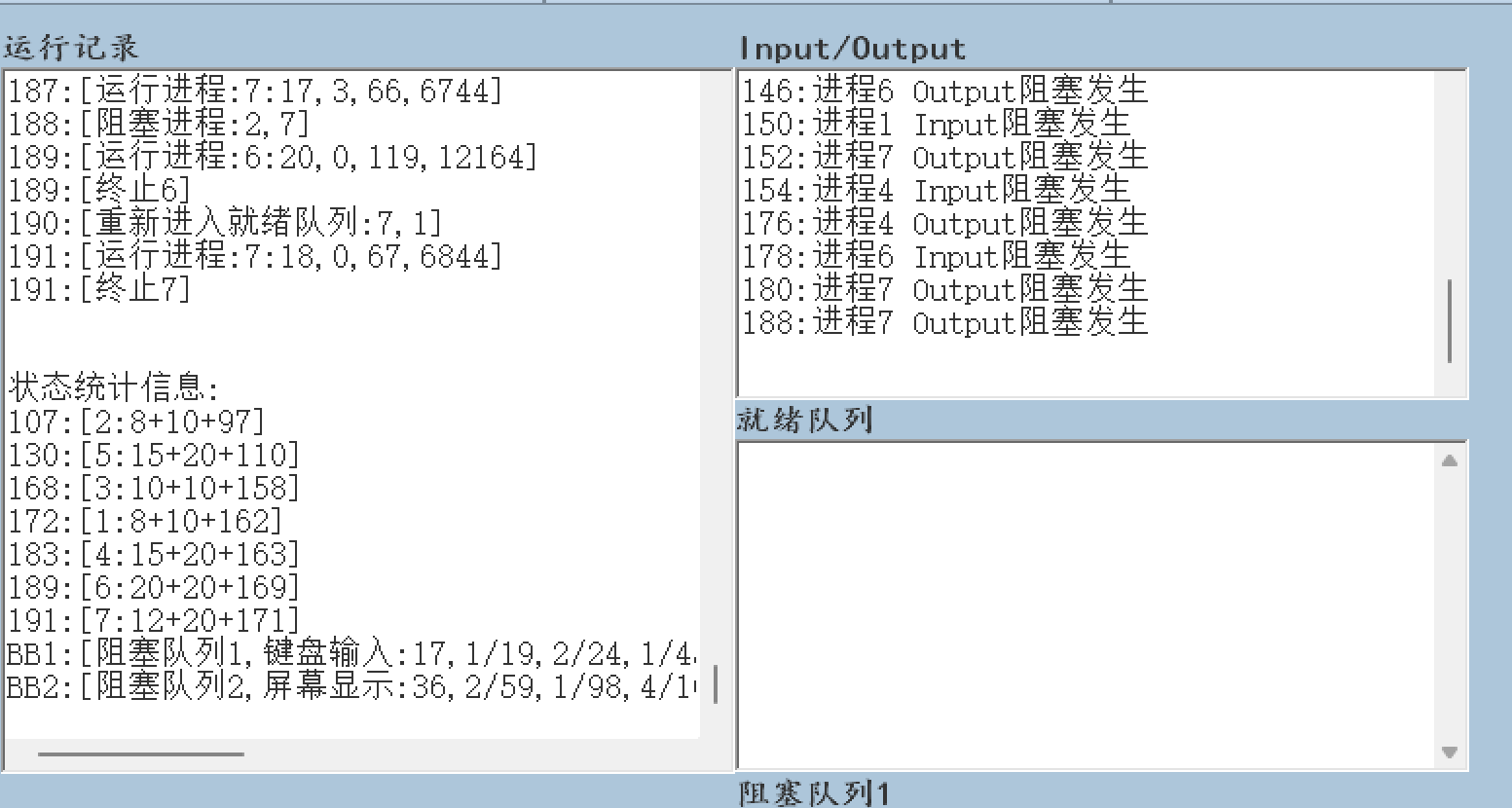
若在还没有开始运行时点击按钮，将会出现提示弹窗：



而正常在运行时点击新增作业按钮，则会显示：



在此之后新增的作业正常参与进程调度：



1. 清空了在终端的不必要的输出System.out.println()：



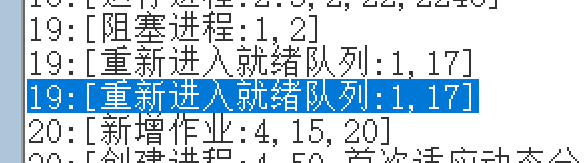
由于这些输入用于编写时调试程序，没有必要保留，于是进行了清除。

5.文件打开部分：

将本系统中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为input1还是input2。（实际上并不是修改了路径字符串，只是在两类字符串值之间进行选择，有错误的可能。）

1. 遇到的问题与解决方式：

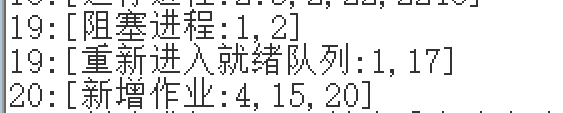
2.对于信息重复输出的问题：



当阻塞的进程时间慢后重新进入就绪队列时，会出现两个重复的“重新进入就绪队列”。

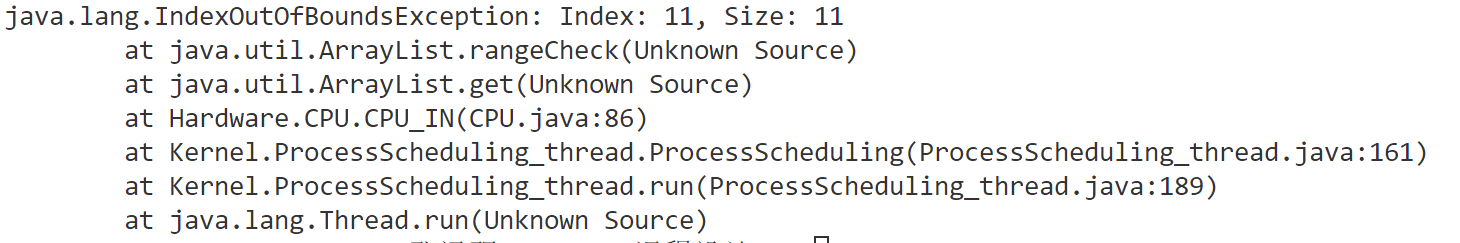
解决方法：

经查找后发现并非调度问题，进行排错之后，发觉是PCB重新进入就绪队列时，在离开阻塞队列时均输出一次该信息，注销其中一个后输出正常：



3.数组越界报错：

当进程的输入输出阻塞为进程的最后一个指令时进程无法正常撤销的问题：

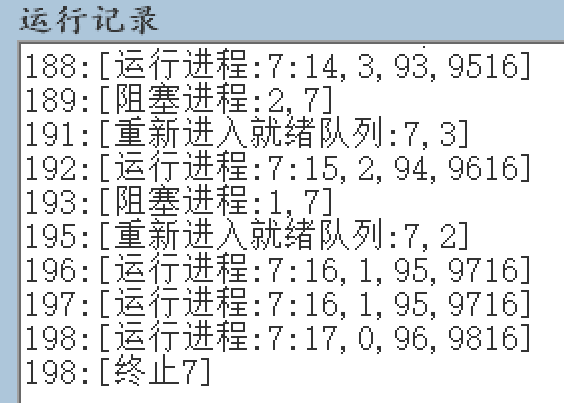


解决方法：

在进程分配CPU时进行指令数量判断，当指令寄存器中指令序号大于进程所含的指令数量，则判断为数组越界，对这个PCB（进程）进行撤销原语的执行。

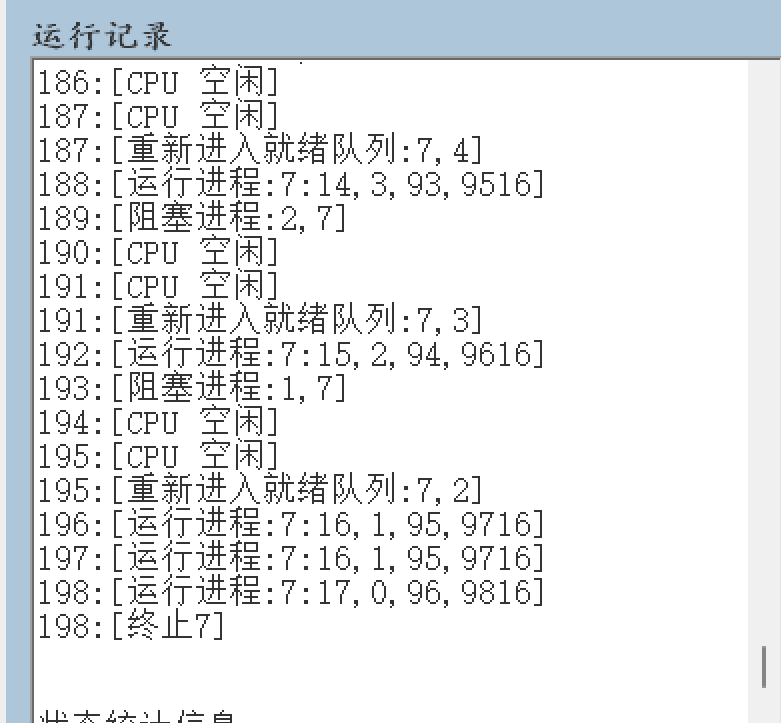
改进后不再出现报错。

4.对于自定义的随机作业，出现阻塞队列非空、CPU空闲但不输出空闲信息：“10:[CPU 空闲]”的情况：



查找以后发现是CPU空闲的判断条件太严苛，修改一些代码使其符合判断条件即可。

改进后：



5.文件打开部分：

将本系统中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为input1还是input2时，对选择的jobs-input文件所属的文件夹名进行判断if(chooser.getSelectedFile().getParentFile().getName()=="input1")满足条件，却没有实现对应分支代码的功能。

进行检查后发现“==”符号对于字符串的对比很容易出错，将它修改为if(chooser.getSelectedFile().getParentFile().getName().equals("input1"))后既可以成功判断jobs-input文件所属的文件夹的名字，进而实现对应分支代码的功能。

1. 待解决的问题：
2. 文件打包为.exe的尝试。