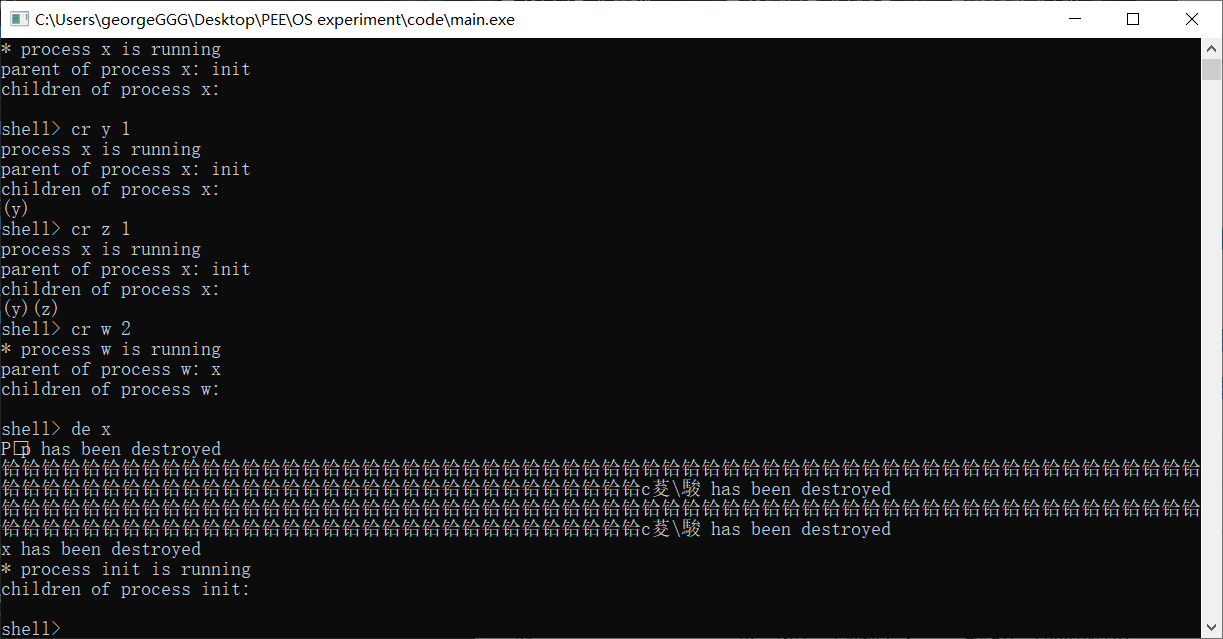
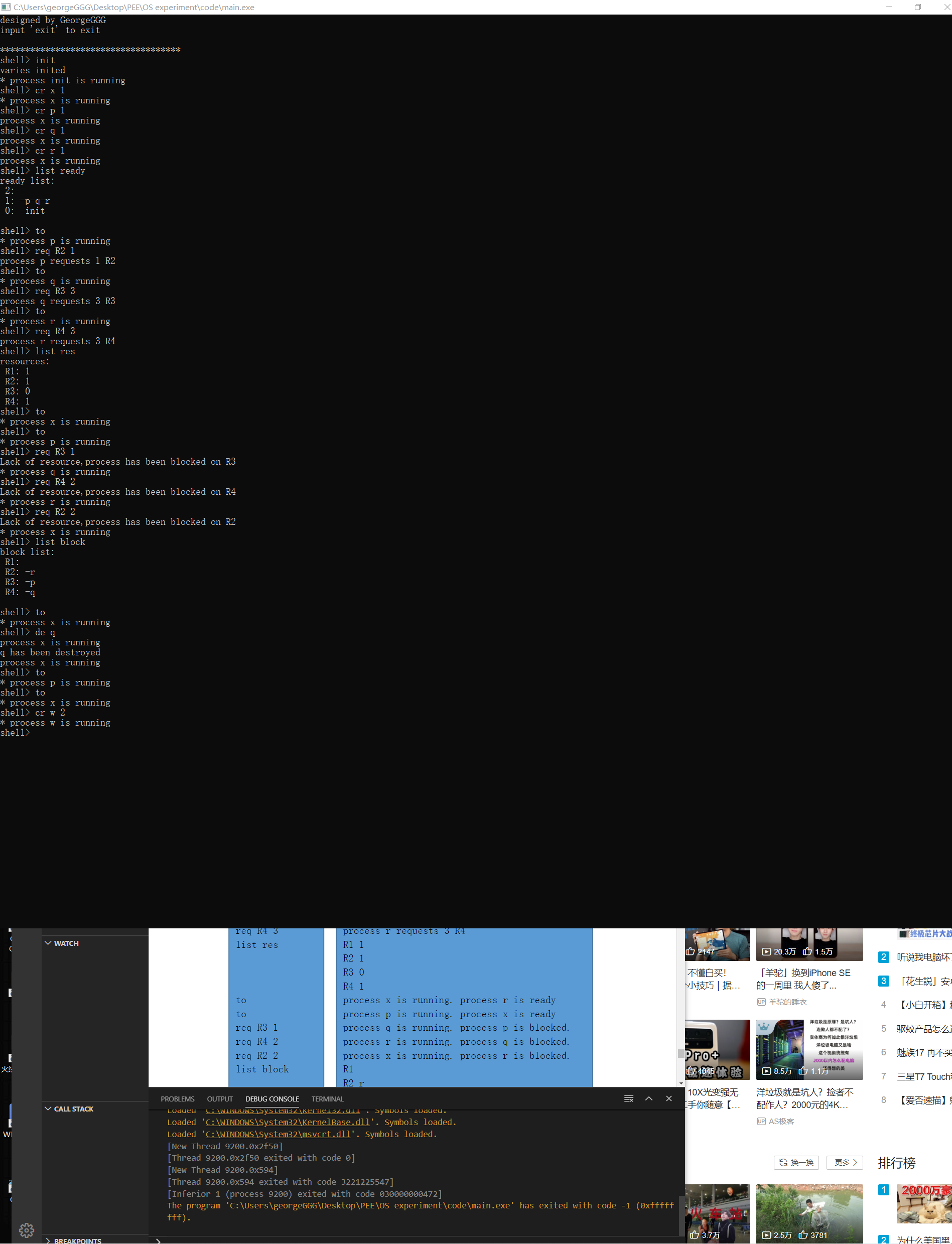
**进度：调试，destroy功能失常—destroy函数中schedule（）调用不当；运行-就绪-阻塞三队列中的inode节点应该是流转的，而不能在每次换队时新建；destroy迭代调用每次删子节点时会删掉一个父进程的子进程记录，而父进程自己在destroy时也删了一个子进程记录，由此造成冲突。**

**基本解决，采用了introduction上的destroy+kill tree双函数法；还有点小问题，输出结果上有点中文乱码**



**成功：**



**设计思路：**

总体设计：

为每个新建进程建立一个索引节点，统一放在一个索引节点表中。

每个进程对应1个pcb和3个inode（comp\_list一个，父进程的子节点列表一个，运行/阻塞/ready队列中一个）

需要编写队列来管理各个队列，所有队列均带头节点，便于处理；不是队列的则不带头节点，如parent和runnning

C语言结构体可以嵌套定义，无需考虑顺序

Test shell：  
strtok函数每遇到一个空格符才分割一次

Scheduler：

关键：preemp（抢占）函数，寻找到read\_list中优先级最高的进程，判断是否优先级高于running进程，若是则抢占。进程自己可以抢占自己（时间片轮转算法内存中只有一个进程时）

资源管理：

假定每次只申请一种资源，则不需要全局的阻塞队列，只需要每个资源管理自己的阻塞队列。

只有running process才能请求或释放资源