|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 题 | 2\_4模拟死锁检测算法  1． 输入：  "资源分配表"文件，每一行包含资源编号、进程编号两项（均用整数表示，并用空格分隔开），记录资源分配给了哪个进程。  "进程等待表"文件，每一行包含进程编号、资源编号两项（均用整数表示，并用空格分隔开），记录进程正在等待哪个资源。  下面是一个示例：  资源分配表：1 1 2 2 3 3  进程等待表：1 2 2 3 3 1  2． 处理要求：  程序运行时，首先提示"请输入资源分配表文件的文件名："；再提示"请输入进程等待表文件的文件名："。  输入两个文件名后，程序将读入两个文件中的有关数据，并按照死锁检测算法进行检测。  3． 输出要求：  第一行输出检测结果：有死锁 或 无死锁。  第二行输出进程循环等待队列，即进程编号（如果有死锁）。  4． 文件名约定  提交的源程序名字：resourceXXX.c或者resourceXXX.cpp（依据所用语言确定）  输入文件名字：可由用户指定; 结果输出到resultXXX.txt中 其中：XXX为账号。  5． 死锁检测算法：当任一进程Pj申请一个已被其他进程占用的资源ri时，进行死锁检测。检测算法通过反复查找进程等待表和资源分配表，  来确定进程Pj对资源ri的请求是否导致形成环路，若是，便确定出现死锁。  6． 测试说明：测试教师将事先准备好一组文件（格式为\*.txt），从中为每个程序随机指定一至三个作为输入文件  （被测试者需从键盘输入指定文件的文件名），并查看程序输出结果。 | |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |