面包商算法：

do｛

choosing[i]=true;

number[i]=max(number[0],number[1],…,number[n-1])+1;

choosing[i]=false;

for(j=0;j<n;j++){

while(choosing[j]);

while((number[j]!=0)&&((number[j],j)<(number[i],i)));

}

critical section;

number[i]=0;

remainder section;

}

while(1);

分析三条红色语句作用：

1.choosing[i]=true choosing[i]=true是标志新进来的进程还没取号排队，置为true表示未取号。如果去掉就不能判断i进程是否是还没取过号码

2.choosing[i]=false 在执行第二条语句后，choosing［i］的值被置为false，标志已经取过自己的号码。如果去掉就不能判断i进程是否取过了号码

3.while(choosing[j]) for循环筛选当前所有排队进程中号码最小的，如果i不是，那么就会在for的最后一句while中等待；若i为所有排队进程中最小的则会跳出for循环进入临界区。这个while循环是为了防止在j还未取完自己的号码就进入第二个while与i比较号码的大小。如果去掉就可能出现以下情况：当前的j在取号过程中发生了中断还未取完号码而把choosing［j］置为false，就被拿去与i的号码比较，那么i就会认为j的一定大于等于自己（初始值为0），这种假象会使i进程不能正常进入临界区。