# 多个缓冲区的进程问题

## 吃水果问题

问题描述：

桌上有一个盘子，每次只能放一个水果，爸爸专向盘中放苹果，妈妈专向盘中放橘子，儿子专等吃盘里的橘子，女儿专等吃盘里的苹果。

代码描述：

plate ← l, apple ← 0, orange ← 0

father{ △爸爸进程

while (1)

准备一个苹果

P(plate) △互斥向盘中取、放水果

把苹果放到盘子里

V(apple) △允许取苹果

end

}

mother{ △妈妈进程

while(1)

准备一个橘子

P(plate) △互斥向盘中取、放水果

把橘子放到盘子里

V(orange) △允许取橘子

end

}

son{ △儿子进程

while(1)

P(orange) △互斥向盘中取橘子

从盘子里拿走一个橘子

V(plate) △允许向盘中取、放水果

吃橘子

end

}

daughter { △女儿进程

while(1)

P(apple) △互斥向盘中取苹果

从盘子里拿走一个苹果

V(plate) △运行向盘中取、放水果

吃苹果

end

}

## 生产者消费者问题

问题描述：

假设缓冲区大小为10，生产者、消费者若干。只要缓冲区池未满，生产者便可以将消息送入缓冲区池，只要缓冲区池未空，消费者便可以从缓冲区中取走一个消息。

代码描述：

Items←0 △缓冲区可使用的资源数

space←10 △缓冲区可使用的资源数

mutex←1 △互斥锁

in=0 △第一个资源

out=0 △最后一个资源

item buf[10] = {NULL} △buf[10]为缓冲区

producer{ △生产者

while(true)

P(space) △等待缓冲区有空闲位置

P(mutex) △保证在product时不会有其他进程访问缓冲区

△product

将新资源放到buf[in]位置

in←(in+1)%10

V(mutex)

V(items)

end

}

consumer{ △消费者

while(true)

P(items) △等待缓冲区有资源可以使用

P(mutex) △保证在consume时不会有其他线程访问缓冲区

△consume

将buf[out]位置的资源取走

out←(out+1)%10

V(mutex)

V(space)

end

}

## 哲学家就餐问题

问题描述：

有五个哲学家围坐在一张圆桌旁就餐，圆桌上有五个碗和五只筷子，他们的生活方式就是交替地进行思考和进餐。平时每个哲学家独立思考问题，饥饿时便试图分别取其左右两侧的筷子，只有两只筷子都拿到后才能进餐；进餐完毕后应立即放下筷子，然后继续思考问题。

代码描述：

chop[5]

semaphore take

main{

chop[5]←{1,1,1,1,1}

take←4

parbegin(Philosopher(i)(i=0...4)) △给哲学家标号

}

Philosopher(i)

{

do{

thinking

wait(take) △请求拿筷子

请求左手边的筷子

请求右手边的筷子

eating

放下右手边的筷子

放下左手边的筷子

signal(take)

}while(1)

}