**作业1：多个缓冲区的进程同步互斥问题**

**问题1：N个共享缓冲区的读写问题：**

1. 多个CP和多个IOP共享N个缓冲区，缓冲区即为临界资源，因此存在进程互斥。而CP进程不断往缓冲区里送出去，而IO进程不断从缓冲区里取数据，二者需要告诉对方缓冲区里是否有数据，因此存在进程同步。这是一个进程互斥以及进程同步共存的问题。
2. CP和IOP为此问题的两个角色，设置计算进程（写入进程）CP，设置打印进程IOP（读取进程）
3. 设置信号量empty表示空缓冲区的数目，信号量full表示已用缓冲区的数目，设置互斥信号量mutex，假设初始时mutex = 1，empty + full = N
4. 解决该问题的伪代码如下：

def CP():

while True:

calculate to product a data

P(empty)

P(mutex)

put data into Buffer[in]

in = in + 1

V(mutex)

P(full)

def IOP():

while True:

P(full)

P(mutex)

get data from Buffer[out]

out = out + 1

V(mutex)

V(empty)

print data

if \_\_name\_\_ == ‘\_\_main\_\_’:

Semaphore empty = N, full =0

Buffer = list(range(5))

in = 0

out = 0

CP()

IOP()

**问题2：盘中可放N个水果的吃水果问题**

1. 爸爸、女儿、儿子三人共同操作一个果盘，果盘是临界资源，该问题存在进程互斥，而三人在操作果盘的过程中，要告诉对方果盘是否为空，该问题存在进程同步。这是一个进程同步和进程互斥共存的问题
2. 爸爸、女儿、儿子是此问题的是三个角色，父亲操作函数Father，儿子操作函数Son，女儿操作函数Daughter
3. 设置互斥信号量mutex，设置信号量rest表示盘子中还能放多少水果，设置信号量apple表示苹果的个数，设置信号量orange表示橘子的个数。rest + apple + orange = N
4. 解决该问题的伪代码如下：

def Father():

while True:

P(rest)

P(mutex)

put fruit in dish

V(mutex)

if fruit is orange:

V(orange)

else:

V(apple)

def Son():

while True:

P(orange)

P(mutex)

get orange from dish

V(mutex)

V(rest)

eat orange

def Daughter():

while True:

P(apple)

P(mutex)

get apple from dish

V(mutex)

V(rest)

eat apple

if \_\_name\_\_ == ‘\_\_main\_\_’:

rest = N, orange = 0, apple = 0

Father()

Son()

Daughter()