

实验名称 苹果橘子问题 日期 2023.11.30 指导教师 赵晓平

专业信息安安全年级班级 21 奇安信 姓名 朱宸扬 学号 202183760012

一. 实验目的

熟悉 PV 操作

二. 实验内容

模拟苹果橘子问题

三. 实验原理

一个盘子， $\text{mutex}=1$

父母分别放苹果和橘子

儿女分别吃苹果和橘子

对应 $\text{apple}=1$ $\text{orange}=1$

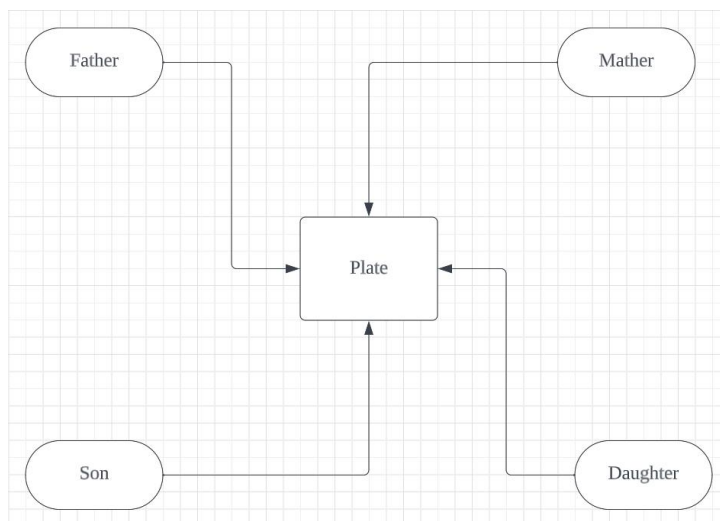
四. 实验设计及编码

1. 模块分析

父母儿女

父放苹果，母放橘子，儿吃苹果，女吃橘子

2. 流程图



3. 代码实现

```
import threading
```

```

import time
import random

mutex = threading.Semaphore(1)
full_empty = threading.Lock()
not_full = threading.Condition(full_empty)
no_apple = threading.Condition(full_empty)
no_orange = threading.Condition(full_empty)

list = []
global count
count=0
def father():
    global count
    while(1):
        with not_full:
            while list and list[0] is not None:
                print("盘子满了，父亲阻塞")
                not_full.wait()
            mutex.acquire()
            print("父亲放苹果")
            list.append("苹果")
            print(list)
            count+=1
            time.sleep(random.randint(1,2))
            no_apple.notify()
            mutex.release()
            time.sleep(random.randint(1,2))

def mother():
    global count
    while(1):
        with not_full:
            while list and list[0] is not None:

```

```

        print("盘子满了，母亲阻塞")
        not_full.wait()
    mutex.acquire()
    print("母亲放橘子")
    list.append("橘子")
    print(list)
    count+=1
    time.sleep(random.randint(1,2))
    no_orange.notify()
    mutex.release()
    time.sleep(random.randint(1,2))

```

```

def son():
    global count
    while(1):
        with no_apple:
            while list==[] or (list[0] is None or list[0] != '苹果
'):
                print("盘子无苹果，儿子阻塞")
                no_apple.wait()
            print("儿子吃苹果")
            list.pop()
            print(list)

            time.sleep(random.randint(1,2))
            not_full.notify()
            time.sleep(random.randint(1,2))

```

```

def daughter():
    global count
    while(1):
        with no_orange:
            while list==[] or (list[0] is None or list[0] != '橘子

```

```

    '):

        print("盘子无橘子，女儿阻塞")
        no_orange.wait()
        print("女儿吃橘子")
        list.pop()
        print(list)

        time.sleep(random.randint(1,2))
        not_full.notify()
        time.sleep(random.randint(1,2))

if __name__ == '__main__':

    # 创建两个生产者线程和两个消费者线程
    father = threading.Thread(target=father)
    mother = threading.Thread(target=mother)
    son = threading.Thread(target=son)
    daughter = threading.Thread(target=daughter)
    # 启动所有线程
    father.start()

    mother.start()
    son.start()
    daughter.start()

```

4. 结果及其相关分析（结果必须是图示）

父亲放苹果
['苹果']
盘子满了，母亲阻塞
儿子吃苹果
[]
盘子无橘子，女儿阻塞
母亲放橘子
['橘子']
盘子满了，父亲阻塞
女儿吃橘子
[]
盘子无苹果，儿子阻塞
母亲放橘子
['橘子']
盘子满了，父亲阻塞
女儿吃橘子
[]
母亲放橘子
['橘子']
盘子满了，父亲阻塞
女儿吃橘子
[]
母亲放橘子
['橘子']
盘子满了，父亲阻塞
女儿吃橘子
[]
母亲放橘子
['橘子']
盘子满了，父亲阻塞
女儿吃橘子
[]

五. 实验小结

加深了对 PV 操作的理解