南京信息工程大学 实验（实习）报告

实验名称 消费者生产者问题 日期 2023.11.30指导教师 赵晓平

专业信息安全年级班级 21奇安信姓名朱宸扬学号 202183760012

1. 实验目的

深入理解生产者消费者问题

1. 实验内容

模拟生产者消费者问题

1. 实验原理

生产者消费者问题

1. 实验设计及编码
2. 模块分析

生产者，消费者，资源池

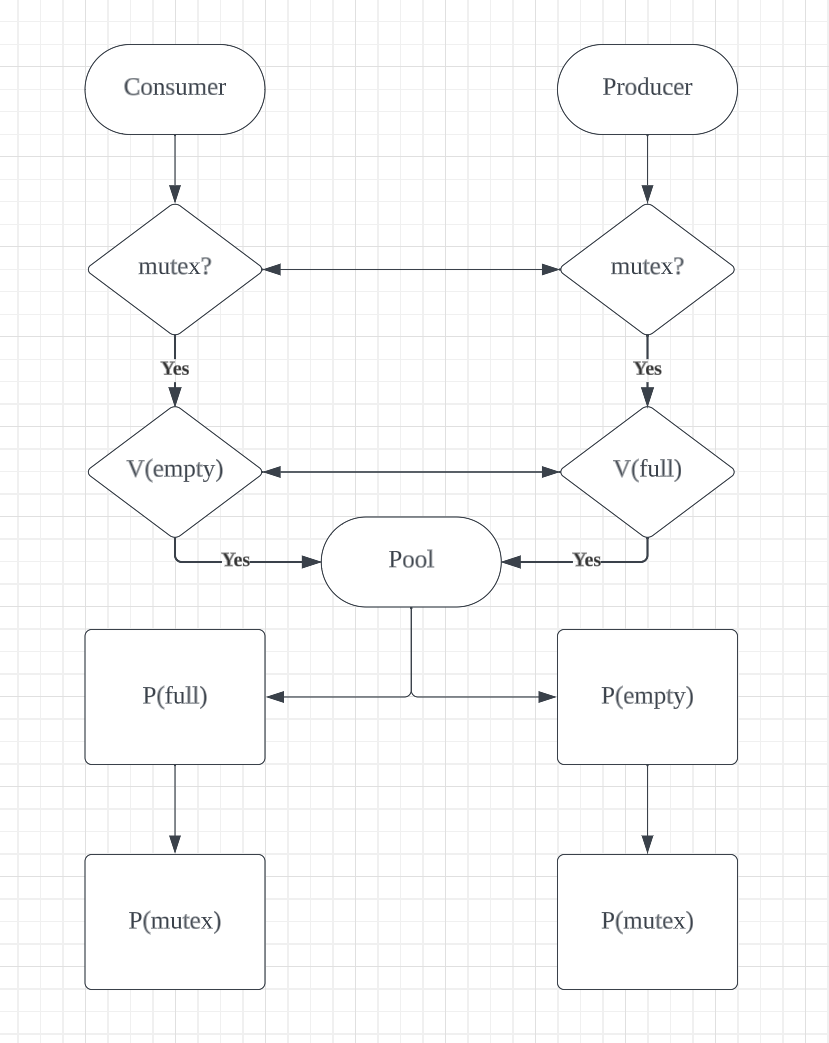
同时只能有一个在使用资源池

无资源时，消费者不能进入资源池

资源满时，生产者不能进入资源池

Mutex=1 empty=0 full=8

1. 流程图



1. 代码实现

import threading

import time

import random

mutex = threading.Semaphore(1)

full\_empty = threading.Lock()

not\_full = threading.Condition(full\_empty)

not\_empty = threading.Condition(full\_empty)

global list

list = [0 for x in range(8)]

global count

count = 0

global pindex

pindex = 0

global cindex

cindex = 0

def producer():

global count,pindex

ptr = [x for x in range(8)]

while(1):

with not\_full:

while count==8:

print("资源池满了，生产者阻塞")

not\_full.wait()

mutex.acquire()

item = 1

list[pindex] = item

print(list)

pindex = 0 if pindex == 7 else pindex+1

count+=1

# time.sleep(random.randint(1,3))

not\_empty.notify()

mutex.release()

time.sleep(random.randint(1,3))

def consumer():

global count,cindex

ctr = [x for x in range(8)]

while(1):

with not\_empty:

while count==0:

print("资源池空了，消费者阻塞")

not\_empty.wait()

mutex.acquire()

item = 0

list[cindex] = item

print(list)

cindex = 0 if cindex == 7 else cindex+1

count -= 1

# time.sleep(random.randint(1,3))

not\_full.notify()

mutex.release()

time.sleep(random.randint(1,3))

if \_\_name\_\_ == '\_\_main\_\_':

# 创建两个生产者线程和两个消费者线程

producers = [threading.Thread(target=producer) for \_ in range(100)]

consumers = [threading.Thread(target=consumer) for \_ in range(100)]

# 启动所有线程

for \_ in range(100):

producers[\_].start()

for \_ in range(100):

consumers[\_].start()

1. 结果及其相关分析（结果必须是图示）

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 0]

[1, 1, 1, 1, 1, 1, 1, 1]

[0, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 0, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 0, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 0, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 0]

[1, 1, 1, 1, 1, 1, 1, 1]

[0, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 0, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 0, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 0, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 0]

[1, 1, 1, 1, 1, 1, 1, 1]

[0, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 0, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 0, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 0, 0, 1]

[1, 1, 1, 1, 0, 0, 0, 0]

[0, 1, 1, 1, 0, 0, 0, 0]

[0, 0, 1, 1, 0, 0, 0, 0]

[0, 0, 0, 1, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 1, 0, 0, 0]

[0, 0, 0, 0, 1, 1, 0, 0]

[0, 0, 0, 0, 1, 1, 1, 0]

[0, 0, 0, 0, 0, 1, 1, 0]

[0, 0, 0, 0, 0, 1, 1, 1]

[0, 0, 0, 0, 0, 0, 1, 1]

[1, 0, 0, 0, 0, 0, 1, 1]

[1, 1, 0, 0, 0, 0, 1, 1]

[1, 1, 0, 0, 0, 0, 0, 1]

[1, 1, 1, 0, 0, 0, 0, 1]

[1, 1, 1, 1, 0, 0, 0, 1]

[1, 1, 1, 1, 0, 0, 0, 0]

[1, 1, 1, 1, 1, 0, 0, 0]

[0, 1, 1, 1, 1, 0, 0, 0]

[0, 0, 1, 1, 1, 0, 0, 0]

[0, 0, 1, 1, 1, 1, 0, 0]

[0, 0, 1, 1, 1, 1, 1, 0]

[0, 0, 1, 1, 1, 1, 1, 1]

[0, 0, 0, 1, 1, 1, 1, 1]

[1, 0, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 0, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

资源池满了，生产者阻塞

[1, 1, 1, 1, 1, 0, 1, 1]

[1, 1, 1, 1, 1, 0, 0, 1]

[1, 1, 1, 1, 1, 1, 0, 1]

[1, 1, 1, 1, 1, 1, 0, 0]

[1, 1, 1, 1, 1, 1, 1, 0]

[1, 1, 1, 1, 1, 1, 1, 1]

[0, 1, 1, 1, 1, 1, 1, 1]

[0, 0, 1, 1, 1, 1, 1, 1]

[0, 0, 0, 1, 1, 1, 1, 1]

[1, 0, 0, 1, 1, 1, 1, 1]

[1, 0, 0, 0, 1, 1, 1, 1]

[1, 1, 0, 0, 1, 1, 1, 1]

[1, 1, 0, 0, 0, 1, 1, 1]

[1, 1, 1, 0, 0, 1, 1, 1]

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 0, 1, 1]

[1, 1, 1, 1, 1, 0, 0, 1]

[1, 1, 1, 1, 1, 1, 0, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

资源池满了，生产者阻塞

[1, 1, 1, 1, 1, 1, 1, 0]

[1, 1, 1, 1, 1, 1, 1, 1]

资源池满了，生产者阻塞

[0, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

资源池满了，生产者阻塞

资源池满了，生产者阻塞

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 0, 1, 1, 1, 1]

[1, 1, 0, 0, 0, 1, 1, 1]

[1, 1, 0, 0, 0, 0, 1, 1]

[1, 1, 1, 0, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 0, 1, 1]

[1, 1, 1, 1, 0, 0, 0, 1]

[1, 1, 1, 1, 1, 0, 0, 1]

[1, 1, 1, 1, 1, 0, 0, 0]

[0, 1, 1, 1, 1, 0, 0, 0]

[0, 1, 1, 1, 1, 1, 0, 0]

[0, 0, 1, 1, 1, 1, 0, 0]

[0, 0, 1, 1, 1, 1, 1, 0]

[0, 0, 1, 1, 1, 1, 1, 1]

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 0, 1, 1, 1, 1]

[1, 1, 1, 0, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

资源池满了，生产者阻塞

资源池满了，生产者阻塞

[1, 1, 1, 1, 0, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 0, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 0, 1]

[1, 1, 1, 1, 1, 1, 1, 1]

[1, 1, 1, 1, 1, 1, 1, 0]

[0, 1, 1, 1, 1, 1, 1, 0]

[0, 0, 1, 1, 1, 1, 1, 0]

[0, 0, 1, 1, 1, 1, 1, 1]

[1, 0, 1, 1, 1, 1, 1, 1]

[1, 0, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 1, 1, 1, 1, 1]

[1, 1, 0, 0, 1, 1, 1, 1]

[1, 1, 0, 0, 0, 1, 1, 1]

[1, 1, 1, 0, 0, 1, 1, 1]

[1, 1, 1, 0, 0, 0, 1, 1]

[1, 1, 1, 0, 0, 0, 0, 1]

[1, 1, 1, 1, 0, 0, 0, 1]

[1, 1, 1, 1, 0, 0, 0, 0]

1. 实验小结

加深了对生产者消费者模型的理解