N个哲学家就餐问题(管程)

Monitor Eat

Procedure get\_chopsticks():

Var chopsticks[N] = 1, i = 0

Wait(chopsitcks[i])

Wait(chopsticks[(i+1)%N]

就餐

Signal(chopsticks[i])

Signal(chopsticks[(i+1)%N]

思考

def philosopher():

begin

while(True):

eat.get\_chopsticks()

思考

End

若干读者写者共享N个缓冲区的读写问题(管程)

1. 读者优先

class monitor write-read:

var buffer[i], i = 0, readcount = 0, item

condition w

def insert(self, item):

w.wait()

buffer[i] = item

j = (j + 1) % N

w.signal()

def get(self, item):

readcount++

if (reandcount == 1) w.wait()

item = buffer[j]

readcount--

if (readcount == 0) w.signal()

return item

def write():

while(True):

item = produce()

write-read.insert(item)

def read():

while(True):

item = write-read.get()

print(item)

1. 写者优先

class monitor write-read:

var buffer[i], i = 0, readcount = 0, item

condition w, s

def insert(self, item):

s.wait()

w.wait()

buffer[i] = item

j = (j + 1) % N

w.signal()

s.signal()

def get(self, item):

s.wait()

readcount++

if (reandcount == 1) w.wait()

mutex.signal()

s.signal()

item = buffer[j]

readcount--

if (readcount == 0) w.signal()

return item

def write():

while(True):

item = produce()

write-read.insert(item)

def read():

while(True):

item = write-read.get()

print(item)