

مکرور مصالح نلاسیت تولید کننده - مصرف کننده باز هم مازی

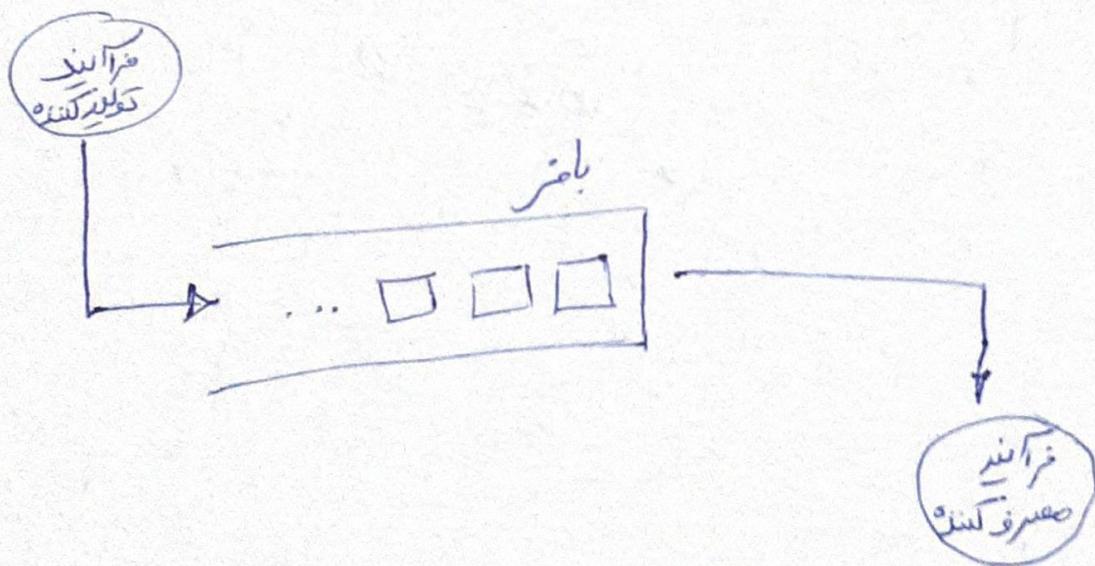
در مصالح تولید کننده - مصرف کننده سرمه حبوبیت زیرا بین این دو نقصی نیست:

۱) فرآیند تولید کننده (Producer Process)

۲) فرآیند مصرف کننده (Consumer Process)

۳) بازتر (Buffer) که می بینند مادر ما نامحدود باشند.

سازی و عمل مصالح برای راه صورت است که فرآیند تولید کننده، آنچه های جدیدی را ایجاد و به باز استرکن افتد و همچنین فرآیند مصرف کننده نیز متناظراً آنچه های جدیدی ایجاد شده را باز استرکن خواهد و اکنون اصفر می کند. مکل زیر سازی برای راه صورت معمور مانند می باشد.



انتظاری بود که در حین درس: متغیرهای مازی (سازگاری اراده)، سرعت زیر همراه  
برقرار باشد:

۱) اگر باز استرکن هم باشد که فرآیند تولید کننده بنا بر آنچه جدیدی را تولید کند

۲) اگر باز استرکن خالی باشد، فرآیند مصرف کننده بنا بر آنچه را مصرف کند

۳) دسترسی به باز استرکن هی بایست بخصوص اکنواری باشند، بین وعنه

که در هر کلمه ممکن است فرآیند باید قادر به رسیدن باز استرکن و تغییر آن

## Producer (void)

{

while (TRUE)

{ item = Produce();

while (cnt == Buffer\_SIZE); /\* do nothing \*/

insert(item);

cnt++;

}

}

## Consumer (void)

{

while (TRUE)

while (cnt == 0); /\* do nothing \*/

item = remove();

cnt--;

}

}

 $\Rightarrow \text{cnt} == 0$ 

اجاری زار خریدار با (و مخواه) اگر همه صحیح هست فالنها کارکنند و اگر خوب نباشد علاوه بر این کارکنند

ترسیب و بطور مجزا لمحه ای  $\text{cnt}$ ،  $\text{cnt}+$ ،  $\text{cnt}-$  را امرازه (متغیرهای دارند) امرازه تغییر می کنند و  $\text{cnt}$  را از اینجا تغییر می کنند. مثلاً  $\text{cnt} = 6$ ،  $\text{cnt} = 5$ ،  $\text{cnt} = 4$  و  $\text{cnt} = 3$  می شود

هر قسم می باشد تولیدکننده - مصرف کننده به ۳ سماور زیر نیاز است:

۱- سماور `empty`: این سماور بمنظور رعایت سُرمه احصار مفهوم (رسانیده) باز فعال است (`mutext.count=1`) و نتیجه این است و مقدار اولیه آن ۰ است (عنوان `empty`).

۲- سماور `full`: سماور بر اساس فعل تعداد خانه های پر باز است و مقدار اولیه آن ۰ می باشد (عنوان `full.count=0`).

۳- سماور `empty`: سماوری بر اساس فعل تعداد خانه های خالی باز است و مقدار اولیه آن ۰ می باشد (حاکمیت بر این تعداد خانه ها باز کافی نیست).

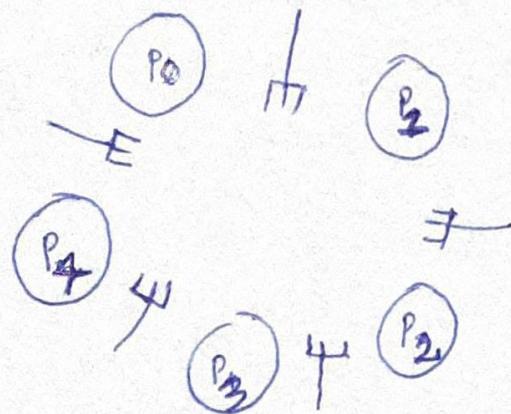
بنابراین رآسیز هار توکیدکننده و معروف کننده را در میان بصورت زیر آنفراست کرد:

producer (void)	consumer (void)
<pre> { int item; while (TRUE) {     item=produce();     wait(empty);     wait(mutex);     insert(item);     signal(mutex); } signal(full);         </pre>	<pre> { int item; while (TRUE) {     wait(full);     wait(mutex);     item=remove();     signal(mutex);     signal(empty); } consume(item);         </pre>

۲۶) تمرین: اگر در مسالم توکید کننده معرفت رفتگی سود، براست از معرفت  
در تلفّر رفتگی سود در بایا، حذف چه مورد؟ معرفت empty دیگر نباشد میتوان در نتیجه  
رسور (wait(empty)) در فرآیند توکید کنند و دستور (join(empty)) در فرآیند معرفت کنند  
حذف خواهد شد.

### مسئله عذاخوان فیلسوف‌ها (Dining Philosophers Problem)

پنج فیلسوف هر یک معیز دایره از نسبت آن‌ها فیلسوف‌یک سُقا به همه کاروئی دارد. بین هم  
جفت از سُقاها چهل قلمرو دارد.



هر فیلسوف هر خود را از دو چنگال (fork) مرضی سُقا اسقاطه می‌کند. زیرا هر فیلسوف از  
دو دوره مختار بخورد (eat) و فکر کردن (think) سُقیل می‌باشد.  
از ماتریک هر فیلسوف گرسنگی سود، سعی کند دو چنگال سمت چپ و راست خود را  
بردارد. اگر موفق شد، برای دسر غذای خود رسید چنگال‌های را زمین می‌گزارد و به  
قدرت اقامی (هد) مسالم تغزیه فیلسوفان علاوه بر انتصار صفت‌گال را در فرآیند  
دو فیلسوف نه تنها از دو چنگال اسقاطه می‌شوند، بلکه باست جو (سبوی) بحسب درستگی  
نیز می‌باشد.

راهنمایی: بجز این مسائل از سمافورهای زیرینه می‌شود:

۱- سمافور room: این سمافور را کان است بر اجازه ورود به سیمای زنجیر تقریباً

روابط در مساحت خنکال دارد. عبارت آن از چهار منیلووف

لمسه باشد، حداقل یک نفر به رو خنکال (سترس) خواهد داشت و لذا این

روابط چهارین سمت را خواهیم داشت. مقدار او شیوه سفارشی این سمافور  
بازمیر ۴ است ( $room\_count = 4$ ).

۲- آکاره از سمافورها fork: بارهای از خنکال های موجود روی صیزی سمافور

در نظر رفته شده است. مقدار او هر سمافور را در نظر رفته شده است

( $fork[i].count = 1$ ).

هر منیلووف ابتدا خنکال خوب و سپس خنکال راست را برعی دارد و بعد از تقدیمی صیزی سمافور

دو خنکال را بر اساس کارهای داشت را روی صیزی نگذارد و سپس می تواند استفاده کند.

نیاین فرآیندهای مربوط به منیلووف را می توان به عنوان تأثیر در نظر رفته:

Semaphore room = 4;

Semaphore fork [5] = 1;

Void Philosopher(int i)

{  
    while (TRUE)

        {  
            think();

            wait(room);

```

wait (fork[i]);
wait (fork[(i+1)%5]);
eat();
signal (fork[(i+1)%5]);
signal (fork[i]);
signal (room);
}
}

```

۳۸

مسئله در راه حل فوق بار اطمینان از عدم خذاری بسیار ساخته شده است، ساخته root را غیر کاربردی می‌دانیم بلکه برای مقادیر فوکوس نیز بسیار صبور است. لیکن راه حل رسیر برای صیلیسی از حین خذار در ترتیب زیر داشت:

کاربر state درین راه حل وعفت خارق مخصوص را به صورت زیر صورت دارد.

۱ آر با state[i]=0 بار بین عناست کر مخصوص آنام در حال فقره است،  
۲ آر با state[i]=1 بار بین عناست کر مخصوص آنام در حال رست است،  
۳ آر با state[i]=2 با استیجین عناست کر مخصوص آنام در حال خوار است.

ویر فیلسوف هنگامی دستیح نمی‌شود از مخصوصات خود بیرون نمی‌گذرد با اینکه باعث خود شدن باشد.

Semaphore mutex=1;

Semaphore S[5]={0};

int state[5];

void philosopher (int i)

```
while (TRUE)
{
    think();
    take_forks(i);
    eat();
    put_forks(i);
}
```

```
void take_forks(int i)
```

```
{
    wait(mutex);
    state[i] = 1;
    test(i);
    signal(mutex);
}
```

```
void put_forks(int i)
```

```
{
    wait(mutex);
    state[i] = 0;
    test((i-1)%5); /* Left Philosopher */
    test((i+1)%5); /* Right Philosopher */
    signal(mutex);
}
```

void test(int i)

```
{  
    if (state[i] == 1 and state[(i-1) % 5] != 2 and  
        state[(i+1) % 5] != 2)  
    {  
        state[i] = 2;  
        signal(S[i]);  
    }  
}
```

سؤال: تحلیل لست دراوه بن سبے بر راه میں اتفاق نہ ہو سکدے؟

راہنمائی: حالت مرنسنے + باغ + take\_fork + تابع سنت

? مرنسد چند ہے؟  $P_i$

$P_{i-1} \rightarrow$  مرنسد

$P_{i-2} \leftarrow$  مرنسد

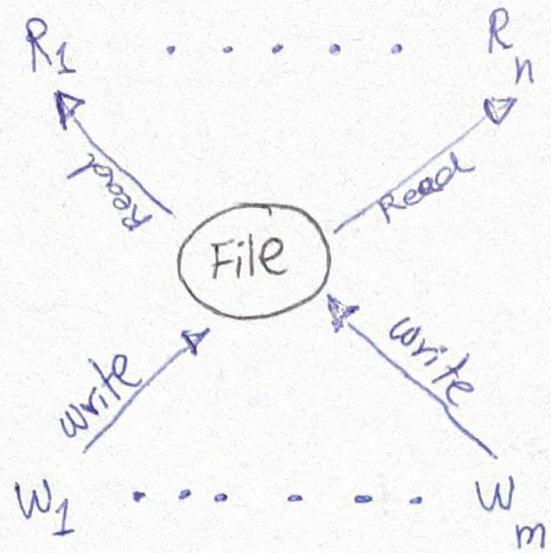
۱۹ مطالعہ خواندن گان و نویسیدن گان (Readers-Writers Problem)

فرض کیا کہ کنڈ یک فائل (رایانہ پر دار) میں تعداد از فرآندہما بر اسٹر آر گزانتر سرو  
اصلت بر تو سیر فرآندہا میں خواندن کے ساتھ زیر ایویو ہے  
خواندن و نویسیدن فائل میں تو سیندہ

۱ ہر آر از خواندن گان میں تولید ہو سو رے ہمیں از قابل بخواہندہ

۲ در ہر لمحہ زصال تنہائی فرآندہ میں تولید در این فائل بخیو سید

۳ ہتھاچی کہ نویسیدن اور دحال نویسن اس سے، همیشہ خواندن کو تولید فائل را  
لخواہندہ



برای این مساله سه بُری به الوبیت‌های ریزن در فضای خواندن و نوشتگان می‌توان کامل شد، بحالت  
وصور دارد:

حالات ۱ = الوبیت بر خواندنگان است، بین مخفی بر تازه‌نگیر خوانشها و موجودات  
بر خوانشها اجازه درود (خواندن کامل) می‌دهند و نوشتگان با این مخفیت ممکن.  
حالات ۲ = الوبیت بر نوشتگان می‌باشد.

حالات ۳ = الوبیت را برای خواندنگان و نوشتگان معمولی برخی سرویس و صوره معرفاً راساس  
زبان ریزن تلقائیاً سرویس همچنانجا بهم می‌سوزد.

در ادامه راه حل برای این مساله می‌شود با مرحله‌ای روی حالت ۱ = (الوبیت خواندنگان) می‌پاسد.

**راهنمایی:** مرحله این مساله از سماقورها و مقدارهای زیر کمتر می‌شود:

۱ = سماقور mutex: به منظور تغییر وضعیت سمتار سایر زمینه‌های مربوط به لذار تلقائیانهای خواندن  
فایل (خواندنگان) قوی استفاده ممکن است (مقدار اولیه آن ۰ است).  
۲ = سماقور counter: مقدار اولیه آن ۱ است (counter = 1).

۳ = سماقور w: برای رعایت این مقدار متعادل سهمای نوشتگان بگزینید در قالب صوره زیر اتفاق نماید:  
قرار می‌شود این مقدار اولیه آن ۰ است (w = 0). (w = 1)

۳) مُهارنده ۲۰: ریاندر هر اتفاقاً هر خواص از ماتل است و در واقع یا نیست که رنده ب  
خواصی تغییر می‌کند، برای اتفاقاً هر خواص دارنده و محدود دارنده، الیت هر خواص ماتل  
دارد سیگنال نویسن در ماتل.

با عصر برآوردهای موقع، مراکز هر خواص و نویس را می‌توان معرفی زیر در نظر گرفت:

~~ویرتنگ~~

Semaphore mutex $\leq 1$ ;

Semaphore w=1;

int rc=0;

Writer (void)

{ while (TRUE)

{ wait(w);

writing();

signal(w);

}

Reader (void)

{ while (TRUE)

{

wait(mutex);

rc = rc + 1;

if (rc == 1) wait(w);

signal(mutex); // Note: +

reading();

احائزه صفر را که حبسه در مسماط / صفر از ماتل

