

`wait(mutex);`

$RC = RC - 1;$

`if (RC == 0) signal(w);`

`signal(mutex);`

}

}

شرح روال نویسندگان: هر فورتی به فرآیندهای قدر در حال استرسی به مالی باشد، اولین نویسندگان با

اجازه دستور `(w) Wait` اجازه دسترسی به مالی (سرازیر کند) (حرایم مقدار w عیم صفر است) است اما به معنی صفر است. این سکافوره اجازه دسترسی سایر نویسندگان را خواهد کرد که فرآیندهای سایر نویسندگان را خواهند کرد. بنابراین در این روال، تا هنگامی که نویسندگان را فعال (دسترسی دارند)، همچنین نویسندگان خواهند بود که توانند به مالی دسترسی سرازیر کنند.

شرح روال خواشته: فرآیند خواشته از ساقوره `W` بر اعمال اختصار صفتیابی مادر دسترسی به مالی به

همنام دسترسی فرآیندهای نویسندگان، از صفتیابی دسترسی W برای هر چیزی تقریباً خواهد بود. از ساقوره `mate` برای اطمینان از تغییر مناسب متغیر W بجزیئی کسر است.

در واقع، دستور `W` اول دستورات عیم وجود نویسندگان مطالعه اجازه ورود فرآیند را فراهم می کند و آن رقم برسی کند که آن خواشته مطالعه دسترسی وجود دارد، در صورت وجود فرآیند نویسندگان مطالعه دارد، دستور

نکته: در این حل باید سمت اختصار صفتیابی و سپریفت رعایتی سود امانتی داشت

انقلابی دور از آن جاییست برای فرآیندهای خواشته هست که فرآیندهای نویسندگان عیم

کامل نباشند، رعایت نظر مورد را فعال دسترسی وجود دارد.

همانطور که در بخش های قبلی صحنه اده کردیم، مسافرها مکانیزمهای محدود را فت می‌دانند. مساله همگام سازی محدود با اینحال استفاده نامناسب آن سبب ایجاد خطا شده و مسأله ایجاد خطا شده و رفع آن مسئله است. علت سخنیه مسأله خطا آن است که بررسی محدود بودن تراکنشها، نتوال کردن ایجاد خطا شده و از عیان محدود محدود نماینده بود ایجاد پیدا کردن اجزای ایجاد رساندن.

نماینده برای نمی‌رسیم که باست دقت زیادی در برگاردن مسافرها در برنامه را نمایند. از هرچه روش توزیعه هندگان آنکه برای نماینده مسئله است از منطق پیش از مساز شروع و مطلع نفروه و باقیتیزه مسئله را بوچنان لغتند.

برخی از خطاها که مسئله است رفته در ادامه آورده شده است:

۱- مسئله است در یک بنا بر ترتیب جریانی تعلیمات (Wait) و

signal() جای به جای شده باشند،

signal (mutex);

critical_section();

Wait(mutex);

در حقیقت واقعیت مسئله است به تصریح محدود محدود که کنفرنسری در بازیهای زیر

محدود قرار داشته باشند

۳ صکن است درین بیناشر عمیقات (Wait) حایلزین سرده یا همیت (Signal)

Wait(mutex);

:
critical section();

:
wait(mutex);

در حین وصفیت، فرآیند جاری درین فراخوانی Wait صدور سرد و صکن است اس سند و دلیل خود (یا محبت جاری از فرآیندهای همکار حذف) سود.

۴ صکن است درین بیناشر نیز یا هر دو عمیقات Wait و Signal را حذف نمی‌نماییم.
در حین مراحلی صکن است اکسچار متقابل عالیت سود یا فرآیند به طور دائمی صسطور (سود).

نباید با توجه به محدودهای بیان سرده در بان برافکان رفع داد آن سیار محتمل است، در ادامه سه دلیلی که ساختار همگام ساز ترتیب بالا به نام Monitor را ارائه دهد
کا برافکان کاربر نویس را راهنمایی کرده و رفع دارد حین مسئله ای را در افقان که نه
می‌گفت.

لازم برگردانه بر حفاظ سعادت‌های اندیب ترتیب لئے عامل سیستیمی بانه بیان زدن (الجهاکان
سیستیمی بانه ترتیب زان برای هم‌ساز را نیز دارند)، صانعین رهای حقیقت ترتیب زان برای هم‌سازی
سیستیمی بیانی می‌سونند.

اطلاعات تکمیلی - نوع داره انتزاعی (ADT) (۴۱)

نوع داره انتزاعی (Abstract Data Type) یک صنعت را فنی از منع های راهه ای
است و افقان را فرامهمی کو و رکرسیب معروف اسما ارمغانی بیان می‌کند.

رفتار آنها بیوں میاز به اطلاع از نحو پیاره سازی آن ساختار را رفتار دارد.

مثال ۱ - ADT برای نوع داده مجموعه های اعداد ممکن ("... و این و این ...")

* عملیات را فرض: +، *، -، /، +، -

* عضایت عناصر: > ۰ < ۰ = < ۰ = > ۰ = و = ای

بعد از درنظر گرفتن پیاره سازی برای مجموعه های اعداد دو درین، معلم ۱، معلم ۲ و ...

مثال ۲ - ADT برای سُسَّتَ (stack): * ساختار انتزاعی با سیاست LIFO

، TOP، Push، Pop + معلم +

خانع از پیاره سازی نه، صورت آرایه را لست سونزی پیاره سازی می سودد و ...

مثال ۳ - ADT برای صف (Queue): * ساختار انتزاعی با سیاست FIFO

+ در عمل DEQUEUE، ENQUEUE

خانع از پیاره سازی نه، صورت آرایه را لست سونزی پیاره سازی نموده ...

ADT برای Monitor

۳۴

نوع داده استزاعی برای نوع Monitor معمول هم بخش زیر است،

۱ بخش تعریف متغیرها: این متغیرها دعوهای که بخوبی از نوع

Monitor را شناسی می دهد.

۲ بخش تعریف عملیات، مسائلی که یا چند تابع است که اخراج آنها می باشد

محاسبه روابط دیگر را اخبار ممکن است و روی متغیرها را فعل مانیتور عمل می کنند.

۳ بخش مقدار دهنده: که مانند دارایی آن، بیان است بمنظور مقدار دهنده

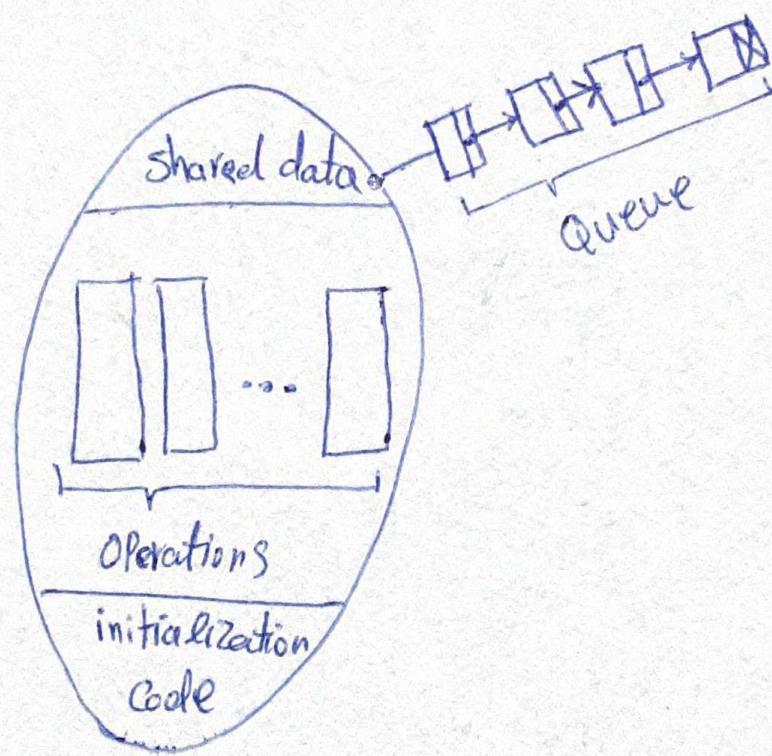
او یعنی بر تعریف راه در مانیتور اسکارا دیگر.

۳۵

ساختار Monitor اطلاعاتی چندین دری را می‌فراند فعال در صاینور رجود طرز
نباید فرآیند همکاری داشته باشد و درین میانور را دارند در صورتی فرآیند فعالی در
صاینور وجود داشته باش، مسدودی شود.

ازم بزرگ است که مخواهد در درین فرآیند به صاینور از طرق فرآیندی پیش از طریق موجود در میانور صورت می‌پذیرد.

جنس های نوع صاینور به صورت معمول در محل زیر قابل مشاهده است



جمع‌بندی: ۱) مخواهد در درین فرآیند به صاینور از طرق فرآیندی پیش از طریق میانور لحاظ صاینور
قابل انجام است،

۲) هر تابعی که در آندر از متغیرها (خلی) آمیخته در داخل صاینور اضافه شود
کند.

۳) متغیرها (خلی) که صاینور را همای تراویز ترسیم نمایند (خلی) صاینور را صورت
(سترسی) می‌گیرند

۴ دریک زمان، تنها یک فرآیند فعل می‌تواند رسانیده حضور داشته باشد.

(۳۴) شاخصار رسانیده تشریح شده در بخش قبل به مامگونه ساختار آنقدر کافی برای حل مسائل همگام ساز را تدارد. در واقع به خود افعال اعتماد نمایند و از دارای یک نیاز این همگام سازی سفارشی را نیز سعیانی لینیم بهین عناوکه در داخل چالنجر متناسب با شرایط فرآیند را حل اجراهی را صدور یا فرآیند صدوری را بدار لینیم.

با این متغیر متغیرهای سُرطان را به بخش تعریف متغیرها اعتماد می‌لینیم.
بر این قدر مسأله از قالب زیر بهره‌مند شویم:

condition conditionName;

نهایتی در روی یک متغیر هر طرف قبل از حذفی محدوده محیط wait و signal است:

* ConditionName.Wait() و بعدین معناست ته مراکن فراخوانده این محیط.

عملق شده و دریک صفت با نام ConditionName

اعتماد می‌شود

* ConditionName.Signal() و بعدین معناست ته می‌شود از فرآیند هایی در عرض

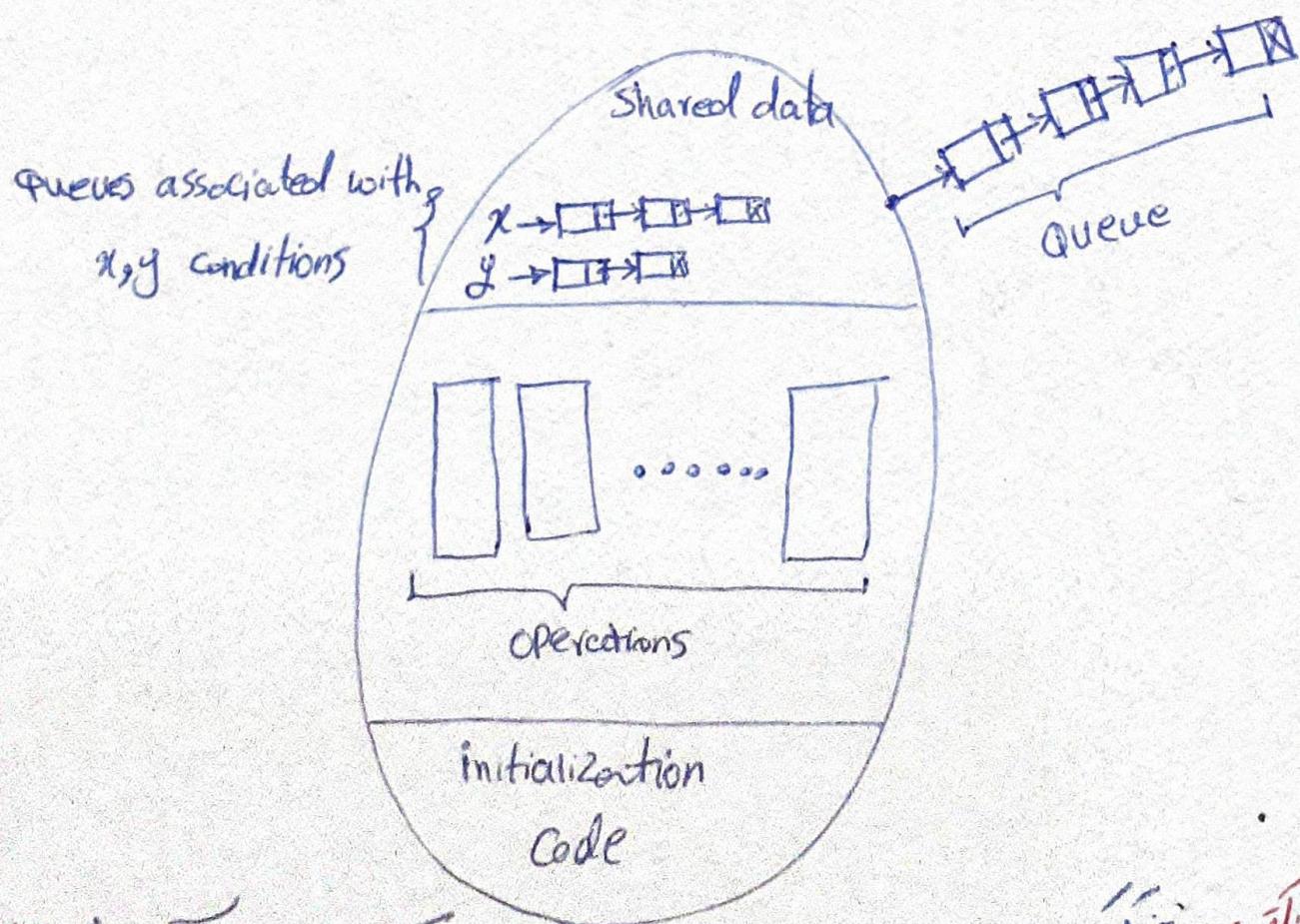
ConditionName صدور شده، بیارگردی و با افزایش حوزه انتشار

می‌شود

نکته: اگر `ConditionName.signal()` فراخوانی شود و هیچ فرآیند مسدود نشده باشد در حقیقت `conditionName` کرد (در مقایسه با دستور `Wait` از اجراء `Wait`). این در حالیست که اجرای دستور `Signal` در ساعورها روی وصفیت متغیرها (سیاره) تأثیر ندارد.

نکته: پس از هر متغیر تعریف شده در صانتور از نوع `condition` باید صفت‌بامهن نام `Shared` یا `Shared` نباشد اگرفا فرآیند مسدود شده برای آن سفره (`condition`) در آن خواهد گرفته.

بخش‌های نوع صانتور با در تظریزش متغیرها) سطری (condition variable) در چهل زیرشان راه دشوه است.



نکته: بر تغییب صفت `Queue` وصف فرآیند `x,y` مقتله کنید. در داخل فرآیند `ما می‌خواهیم`

صفحه اولیه در حالت ایندکس از ماتریس را درست کنید و ماتریس همچنانجیب
گذاشته شود در صورت ایندکس ایجاد نمایند و این دستور را ایجاد کنید و بعنوان مبارز تراجم معمولی
و مجهود در صورت

در صورت ایندکس در صورت

جمع سنبی: شبکه در ساختاری ماتریس ایجاد کنید و بعنوان مبارز در نظر گیری کنید.

monitor monitorName

{

+ Shared Variable declarations */] Variables

function P₁ (...)

{

:

}

:

}

:

:

:

:

:

:

:

:

:

:

function P₂ (...)

{

:

}

:

:

:

:

:

:

:

function P_n (...)

{

:

}

:

:

:

:

:

:

:

Initialization_code (...)

{

:

}

:

:

:

:

:

:

:

functions

initialization

حل مساله تولید کنندۀ - معزوف کنندۀ به کم ماینیور ۱۴۵

برای حل مساله بکار رفته ماینیور، نویی بحرا نی را داخل ماینیور تعریف می کنیم. مینیون
برای بخش هایی در مساله که نیاز به حفاظت سازی وجود دارد، از متغیرهای مطرده
گذاشته شود.

monitor ProducerConsumer

{

Condition full,empty;

integer cnt;

function Produce ~~insert~~(item)

{

if (cnt == BUFFER_SIZE)

full.wait();

~~insert(item);~~

cnt = ~~cnt~~ + 1;

if (cnt == 1)

empty.signal();

}

item function Consume ~~(item)~~

{

if (cnt == 0)

empty.wait();

```
item = remove();
cnt := cnt - 1;
if (cnt == (BUFFER_SIZE - 1))
    full.Signal();
}
return item;
```

Initialization_code()

```
{ cnt = 0;
}
}
```

Producer (void)

```
{ while (TRUE)
{
    item = produce();
}
}
```

ProducerConsumer.produce(item);

```
{ Consumer()
{
    while (TRUE)
{
    item = ProducerConsumer.consume();
}
}
```

item = ProducerConsumer.consume();

PA } } consume(item);