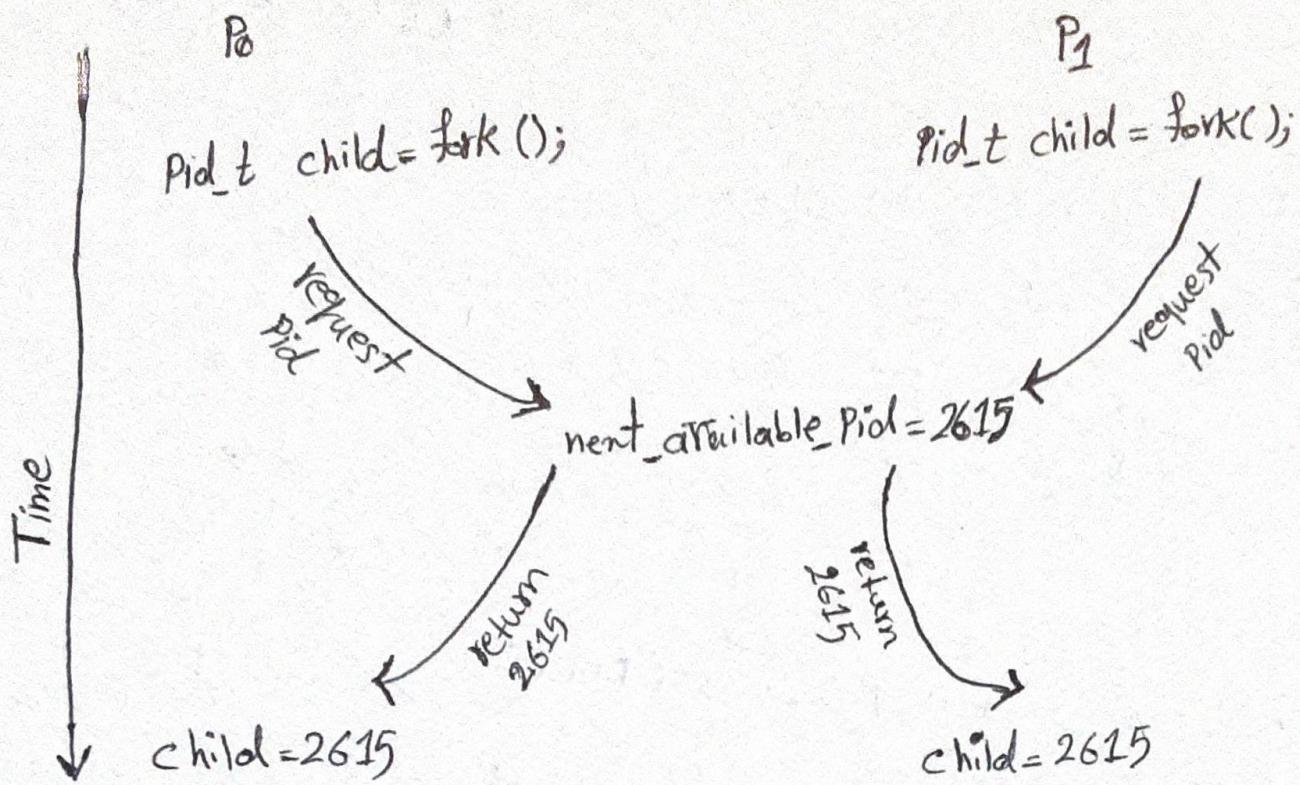


مثال ۵ - دو فرآیند P_1 و P_2 را در نظر بگیر که با استفاده از فراخوان سیستم fork خواهند صباورت به ایجاد فرآیند خرزند نمایند. همانطور ترقباً بیان می‌شود این اسکرپت fork، سیستم می‌بیند که PEP فرآیند خرزند خواهد بود و در درست آن که فرآیند پرگرانده سکونت خواهد بود.

هر چند شرایطی، روی متغیر $next_Variable_Pid$ متغیر سیستم Δ می‌رسی Pid مربوط به فرآیند کامل ریترس نمودی است، شرایط کامبینیشن

(Race condition) رخداد دارد و ممکن است pid کلیان بتوفر کردن مسئله تخصیص داره باشد.



مثال - مزاید رقابت به نظم تخصیص pid بفرآیند فرزند

مثال ۴ - داده ساختارهای مرتبه پر گذاشت تخصیص حافظه لیست فراکندها و مسئل کردن وقفه ها، از جمله داده ساختارهای رشی هستند که اعماق رفراود مزاید رقابت برای آنها وجود دارد. در حقیقت مزاید برنامنویسیان کردن با این طبقیان حاصل نتیجه توقفهای کافی نماید تا سرعت آنها به نحوی انجام می شود که مزاید رقابت درست رفع نشود.

جمع شدی:

بر حفاظت از صفت (یا یا معتبر) داده استوار در مقابل Race condition نیاز است.

است این روش رهنمایی فراکنند درین زمان اعماق دستگاری داده اسرائیل را دارند. برای این منظور سازنده روشی برای هماهنگی (synchronization).

فرآیندها دارند.

در رابطه با فرازگشتهای مارکیپ می‌نمایست با این مساله لنترس زیر برخورد شود:

۱ عدم انحصار متقابل: فرزن لند هر یک از کاربران را درست بدهی منبع رقابت می‌کند. بنابراین در صنعت امراء هر یک از فرازگشتهای میزانی را به منبع مذکور ارسال می‌کند. حین منبع را منبع محراجی (critical Resource) و بخشی (critical section) از بین افاه که از آن استفاده می‌کند را بخش پیغام (critical section) یا نافعه محراجی (critical Region) می‌نویسم.

۲ در حین فرایند می‌باشد مهواره درین زمان تنها یک بخش از فرایند را بخش پیغام می‌فرماید هر دو بخش.

۳ بنبست (Deadlock): در فرازگشتهای P_1 و P_2 و دو منبع محراجی R_1 و R_2 معرفی شده است. بخوبی که هر یک از فرازگشتهای مذکور را در انجام عمل خود بر هر دو منبع نیاز دارد. در حین فرایند، اگر منبع R_2 به P_2 و منبع R_1 به P_1 داده شود، هر یک مختلف رشته منبع خود را نیاز خود را باشود و بنابراین هیچ کدام از آنها منبع خود را در اختیار دارند. در این حالت نیاز فرازگشتهای P_1 و P_2 را در بحث کرد و بخش محراجی خود را انجام نمی‌دهد. بنابراین هر دو فرازگشتهای P_1 و P_2 تا اید متقطع و مردود بر این عیوب محراجی خود می‌مانند.

۴ هرسنگ (Starvation): فرزن لند هر یک از فرازگشتهای P_1 ، P_2 و P_3 می‌تواند با نیازمندی راسترسی به منبع R پیدا کند. هنگامی که P_1 این منبع را در اختیار گیرد، P_2 و P_3 در استقلار آن منبع به تأخیر از اینها می‌سوزند. با خروج P_1 از نافعه محراجی، تکه از P_2 و P_3 با در منبع R ۱۵

را در اختیار گیرند. آن‌ر پ ۳ منبع R این بگیرد و قبل از همایان بنفس بحثانی، معتبراً P ۱ در خواست
R را صادر کند و بعد از همایان P ۳، هم P ۲ اجازه در اختیار رفتن منبع R دارد سو و صفر را
این محمل بین P ۴ و P ۵ ازمه را ببرد، P ۶ به صورت نامحدود از رسترس به منبع R عرضه
می‌نماید و در ترتیب بدان این فرآیند پوله رنسنگ حاره‌ای می‌سازد.

۱۰ تهارست آرسنل ون سبک، هر دو صفحه‌هم به سبب اعمال احتصار مقابله خود با اتفاقات
فرسنگ، برای فرآیند هم‌طور عمل تعریف می‌شود و بین معناس است که فرآیند پر
مقدّس نامعلوم و بقول حد بالا مخصوص منطقه فرآیندهای دیگری می‌نماید.
بن سبک: برای مجموع از فرآیندهای هم‌طور عمل تعریف می‌شود و بین معناس است که
این مجموع فرآیندهای ایده‌استفاده و در نهاده بحثانی خود را ماند.

۱۱ مسئله تاکید بحثانی (critical-region problem)
منطقه کنترلی که شامل یک مجموعه از فرآیندهای P ۱، P ۲، ...، P ۴ موصود
است. هر فرآیند درین مجموعه شامل یک بخش بحثانی است که در آن
فرآیند مکمل است. دادهای را که در افق ایلک فرآیند ریز به استراکتور را
درست آیند یا بروز رسانی کنند. و تراجم صور استفاده در حین پیغامبری
آن است که می‌بینیم دو فرآیند در زمان ملیمان تاکید بحثانی خود را
اجرا نکنند.

مسئله تاکید بحثانی، اراده پر و کل است. فرآیندهایی که می‌توانند حین
استراکتور داره ملیمان ~~مکالمه~~^{پیغام} مخاطب ساز و خالیت های ملیان از
آن استفاده نکنند.

پر و کل هایی که برای حل مسئله تاکید بحثانی اراده می‌سوند، عمدتاً شامل

سی بخش تبرهستند:

۱ بخش ورود (Entry section): هر قرآندر برای ورود به بخش بحرانی خود نیاز است کا مجوز نازم را لسب لند. به این بخش از کد، بخش ورود (Entry section) اطلاق می شود.

۲ بخش بحرانی (critical section): یخنس از کدم بوجو طبیر خراکند اسکا که در آن داره ارم با فرآنفرس سر بر استارگ نداشتند سده را صور دستیابی کا بروزرسانی خواهد داشت.

۳ بخش خروج (Exit section): هر قرآندر برای خروج از ناصیر بحرانی خود نیاز است کا اقدام را انجام دهد. به این بخش از کد، بخش خروج (Exit section) اطلاق می شود.

۴ بخش باقی مانده (Remainder section): مابقی کدهای فرآیند کت بخش باقی مانده در نظر رفته می شود.

بنابرین ساختار کدهای قرآندر به این شکل باقی می بینیم بندهای بین سده در پایان صورت زیر است:

while (TRUE)

{

Entry section

critical section

Exit section

} Remainder section

۱۲ نکته: علت داخل کار در مدارج بخشندهای ورود و خروج به سیستم
آنهاست که در واقع بروگاهی از آن سده بررسال تاحدی بحرانی
بران دو بخش تمثیل دارد.

۱۳ یک راه حل برای مصالح تاحدی بحرانی می باشد سمت زندگی زیر را
ارتفاکند همکاری درست و کار بین فرآیندهای هموندی قرار گیرد؛
۱ انتصار متقابل (mutual exclusion): از بین فرآیندهای که برای
یک منیو تکیان دارای تاحدی بحرانی هستند در هر یک قدر فقط
یک فرآیند مجاز است که در تاحدی بحرانی خود باشد.

۲ پیشرفت (Progress): فرآیندی که فعلاً دقتیم پر تاحدی بحرانی
را ندارد و در تاحدی غیر بحرانی است (رسانی عملیاتی عاری را (اعراض) نماید در تصفیه سری برای درود فرآیندهای ریز پر تاحدی بحرانی سریک
کند (اعکان صفات نداشت باش).

۳ انتظار محدود (Bounded Waiting): می باشد صفت از مان انتظار
فرآیندهای که نیاز به ورود به تاحدی بحرانی خود را دارند، محدود
باشد. به عبارت دیگر، فرآیندهای نیازد حوار رسانی وین بسیار
سزوند.

شاید درست نامعلوم و میتوان حد بالا را مسخن متفاوت
فرآیندهای ریز باشد

بنیت: کابد متفاوت ورود به تاحدی بحرانی خود باشد.

نکته: بروگاه از آن سده بررسال تاحدی بحرانی می باشد مطوف (غیر مقادی)
باشد، خدا!

فرآیندها برای دریافت مجوز و درود به تأیید بحثی (در بخش درود) در حالت
کلی استقلار (برای برآمدگیری) را می توانند متحمل شوند:
۱) استقلار مسُفول (Busy Waiting)
۲) مسدود کردن (Blocking)

روش استقلار مسُفول در این مسئلله اختلاف ~~محدود~~ بردازند است و از این رو
مجموعاً هنگامی استفاده می شود که زمان استقلار کوتاه باشد.

۱۵) روکیر (های) کامپیون انحصار متقابل
برای تحقق انحصار متقابل، چهار روکیر کلی زیر مطروح است:
۱) روکیر نرم افزاری،

۲) با محابیت سخت افزار (با عکس (س سورا لعمل های خاص (PL))

۳) با محابیت سیم عامل (با عکس فراخوان های سیستم خاص)

۴) با محابیت زبان برنامه سازی (با عکس کامپیلر).

۱۶) روکیر (های) نرم افزاری انحصار متقابل
راه حل های نرم افزاری مستقیماً توسط برنامه های استفاده می شوند و مجبور حافظه
استراکتی در این روکیر انتزاعی است. در این راه حل ها از رستورالعمل های خفن
توسط سخت افزار استفاده نمی شود و محابیت از سیم عامل و زبان های پر امپسازی
وجود ندارد. در این راستا در ادامه به دو الگوریتم زیر برداختیم:
۱) الگوریتم دکر (Decker)،
۲) الگوریتم پرسون (Peterson)

نکات که باید در هر این یک پروتکل معمول برای مسائل ناچیز بجزئی در نظر بگیریم:

۱- پروتکل برای هر کسی که لنتیم فقد قسمت Entry section و Exit section است ساختار معرفی شده راسماً هم سر دارد بخش‌ها (ناصیحه مجران و غیر مجران) مربوط به برنامه راست و در هر اینها مراحل و تصریفی در آن قسمت دارند.

۲- ناصیحه مجران و غیر مجران طبق دسته‌بندی برای هم‌ورودی انعام می‌سرد و این نتیجه که در این ساختار پروتکل مرکانه عین ناصیحه مجران را بعنوان ناصیحه مجران در نظر نگیریم لایه‌های مختلف Entry section و Exit section همچنان که این نتیجه نسبت نفر سردار

۳- صفت‌های اولیه تغییرهای لنترنس برای پروتکل می‌باشد که ظور دقیق تعیین سردار و در روال سوچیت در هر این پروتکل تغییر نقص کمی اتفاق نماید.

۴- در کلیه پروتکلهای از این‌روزه نرم‌افزارهای خوب را برآورده که لذاریم که هر یک از افراد این‌ها با سریع غیر قابل اجرا فی سووند.

۵- نحوه انتظار برای ملاقات حیوز و ورود به ناصیحه مجران ترتیب فرآشته‌های تولید و روکاب با استفاده از زیر انعام سرو.

۶- انتظار ممنوع (Busy Waiting) : حلقة مانند.

۷- احیانه مسدود کردن (Blocking) : با استفاده از وقفه.

۸- مسائل حل کردن خود لنترنس داده قشر را سهیل کی ننمی‌بینیم به مسائل حل کردن خود لنترنس که مجران (حمل روی راه استراتژیک) هستند.

۱۷

آلوریتم دکر
آقای دکر ارسن سفیر بود که راه حل زیر افزایش فراشندی برای مساله انعقاد
متقابل ارائه کرد. در ادامه پیچ تلاس اسماں پلر سید به راه حل درست را
صروفی کنیم که از کار آموزشی (زمینه هار مصوب) حاتراهیست است.

آلوریتم دکر - تلاس اول : در این روش از یک متغیر سراسی مشترک به نام turn
استفاده می شود که یک عدد دو رقمی است که را اختیار کند. صفر را اولین متغیر
است. شکل زیر ساختار کلی دو فرآنشت P_0 و P_1 در تلاس اول آلوریتم
دکر را نشان می دهد

int turn=0;

P_0 (void)	P_1 (void)
{	{
while (TRUE)	while (TRUE)
{	{
ENS [while (turn != 0); /*wait*/]	ENS [while (turn != 1); /*wait*/]
CS [critical-section();	CS [critical-section();
EXS [turn = 1;]	EXS [turn = 0;]
NCS [non-critical-section();]	NCS [non-critical-section();]
}	}
}	}

ENS = Entry section, CS = critical section

EXS = Exit section, NCS = Noncritical section

در این آلوریتم، با پیکربندی صحیح شده در بالا، استرا P_0 و P_1 بتوانند وارد ناچشم بصرانی خود شوند. در این حالات P_1 در حلقه while می تواند می ماند که از P_0 از

تا خیر بحرانی خارج شود و turn را با \neq صدرازیر کن. در این صورت معملاً کتوں
فرآید P_1 هم کو اندوارد نا خیر بحرانی شود.

حال زمان آن است که سمت از فندر مربوط به هر راه حل معین برای مساله نا خیر بحرانی
را بررسی کنیم:

بررسی انعصار متقابل: با توجه به پیگیرنیز از سده، و پیگیر انعصار
متفاصل که اینه آنلورن P_1 برقرار است و امکان دارد P_0 و P_1 هم موقت متمایل
(در کنفله) در نا خیر بحرانی خود باشند. حرف اخر P_0 زمانی دارد نا خیر
بحرانی خود میگوید صدرازیر $turn = \emptyset$ ناسد و زمانی این صدرازیر $turn = 1$
لیکن لذتگیر فرآیند P_0 دست در نا خیر بحرانی خودست و فرآیند P_1 امکان حضور
میباشد و اینه تکرار به همینه مسئوال تکراری شود.

بنابراین به هنگام دریافت عجز امکان دارد هر دو بر $turn$
نهاده کنند و هر دو باهم صدرازیر \neq و \neq را پیشته.

Timeline
شل زیر بحرانی (Timeline) (Timeline)
مجه شونه از احراز (Proving) فرآیند P_0 و P_1 را نشان می دهد

