

① زمانبندی CPU (CPU Scheduling)

زمانبندی CPU، پایه و اساس سیم عامل های حین برآمدگار در سطح کلان و حین رزونانسی و حین ترقی در سطح خرد (رندرینگ) می باشد.

در سیم عامل های حین برآمدگار سعی بر اینستاده حداکثر از CPU با این راه حین برنامه بمحورت صریح (لایو اری) برسیلر سوئیچ کرن بین فرآیندها هاست. در جلس سیم هایی می باشد استراتژی برای تعاب فرآیند و سوئیچ به آن باشیم. این موضوع تجتیفی زمانبندی CPU مطرح می شود.

لازم به ذکر است که همه متابع در سیم کامپیوترازی زمانبندی می شوند و این زمانبندی متفقن CPU نیست. با اینحال از آنجاییکه CPU نیکی از متابع اعلی سیم های کامپیوترازی به حساب می آید، زمانبندی CPU در مرکزیت را عمل طراحی سیم عامل مطرح و بررسی می شود.

مقاصد پایه مطبع در زمانبندی CPU به مرزبراسکاتر را دام بر مطلع هر کی از آنها خواهد بود.

= حیرخه انفجار I/O (CPU-I/O Burst cycle)

* CPU scheduler

۳ زمانبندی دون مبنیه (ایمنی) و با مبنیه (عنیکعنی)

(Preemptive and Nonpreemptive scheduling)

۴ توزیع کننده (Dispatcher)

۵ حیرخه انفجار I/O-CPU: وقوعیت اینزبورن زمانبندی CPU (افزاس پیروی و کاهش زمان کاشف) و استفاده از فرآیندها

دارد. بطور لایه لایه فرآیند موارد بین روحاللستا (حیرضه) جایجاًی سرده

- حیرضه اجرای CPU

- حیر خدم انتظار

در افع، اجرای بک فرآیند با بک انفجار CPU آغازی سرد و با رخ دادن بک انفجار I/O برای فرآیند جاری، بک انفجار CPU جدید پر از فرآیند زیر حارث شده و این تراوی تکراری سود. در نهایت نزدیک موقت انتظار CPU بدلی انجام در مقاومت خالق اجرای فرآیند حارث و فرآیند خانق حی باید. همچنان زیر توالت سفری شده در باتارا نشانی دهد

⋮

load store

add store

read from file

CPU Burst

wait for I/O

I/O Burst

store increment

index

write to file

CPU Burst

wait for I/O

I/O Burst

⋮

⋮

نهاد: تراوی جایجاًی انفجارهای I/O, CPU

حدتازمان انفجارهای CPU بطور لسته دار درین ترتیب کامپیوتری اندیزه نمای سود و برهگام پیاره سازی آنلور تتم هارز مابتدی CPU صورت اسقاطه هماری سود

۳۵ نکته: اتفاقهای CPU از فرآیندر برقرار کننده و از ستم کامپیوئری برتر کامپیوئری دستور مفایل است، اما در حالات ممکن اندازه دیری هار انجام شده از اتفاقهای CPU را بگذران در رو روزه کن زیر تقسیم نموده و از اطلاعات آنها در حالت برقراری باشندی بهره ور فرم:

- اتفاقهای CPU طولانی (Long)

- اتفاقهای CPU کوتاه (short).

بنابراین براساس مقادیر بین میان برنامهای را برعهود زیر تقسیم کرد:

- برنامهای I/O Bound: برنامهای هستند که به طور محض شامل تعداد زیادی اتفاق CPU کوتاه می باشند.

- برنامهای CPU Bound: برنامهای هستند که به طور محض شامل تعداد کم اتفاقهای CPU طولانی می باشند.

۳۶ زمانبند CPU Scheduler (CPU scheduler) هنگامی که CPU بیکار (Idle) می شود، سیستم عامل می باشد که فرآیند را از میان آنها انتخاب کند. این انتخاب توسط مولفه ای برنامه زمانبند CPU انجام می شود. در واقع این مولفه از صیان فرآیندهای موجود در حافظه کم آنهاه اجرا می شود، فرآیند انتخاب و CPU را به آن اختصاص می دهد.

نکته: کافیم بود راست که معرف فرآیندهای آنهاه لزوماً از استراتژی اوپری ورود

اوپری خروجی (FIFO) استفاده نظر کند. در ادامه خواهیم دید که با توجه به الگوریتم های زمانبندی، میان آنهاه می تواند به عنوان اتفاقهای زیر پیاده سازی شود:

- یک صفحه محصور با استراتژی FIFO،

- یک صفحه الونت.

- لیک درخت
- لیک لیست سوتندی نامه

با اینحال و در حالات کلر، همه فرآیندهای موجود در صفحه آماده در انتقالار فرستاده
برای اجرا روی CPU هستند. همچنین PCB های مربوط به هر فرآیند،
عفاقس رفیق را شامل می‌شوند

۵) زمانبندی بیان حقیقت (اعماری) و باقیت (غیر اعماری)

تفصیلی مانند امر بروطیه زمانبندی CPU که سرعت زیاد صورت می‌شود:

۱) حقایقی که فرآیند از وقایت اجرا به وصفت انتظار تغییر وغایت
می‌شود. این تغییر وغایت می‌تواند بنا بر رالیه نظری انتظاریک فرآیند
بلای خاتمه فرآیند فرزند و با درخواست I/O را انتقال ببرگشتل آن
پاسد

۲) حقایقی که فرآیند از وقایت اجرا به وغایت آماده می‌شود به
عنوان مثال وقتی که وقفه حارمه می‌شود (می‌شود) تغییر وغایت رفع
مثل آنکه نازه زمانی مجاز بیان اجرا
می‌شود

۳) حقایقی که فرآیند از وقایت انتظار به وصفت آماده سوییج
می‌شود. به عنوان مثال وقتی که درخواست I/O تکمیل می‌شود
غایب وغایتی رفع می‌شود.

۴) حقایقی که فرآیند طیان می‌شود.

۶) حقایقی که زمانبندی که سرعت ۱ و ۲ از موارد طالا صورت می‌شود (۱) می‌تواند هرچه
زمانبندی غیر قبضه‌دار (انتهایی) است. در این نوع زمانبندی، حقایقی که زمان
که فرآیند احتفاظ می‌شود (فرآیند CPU) را که زمانیکه خاتمه ایده به حالت انتظار
تغییر وغایت دهد، در احتیاط دارد.

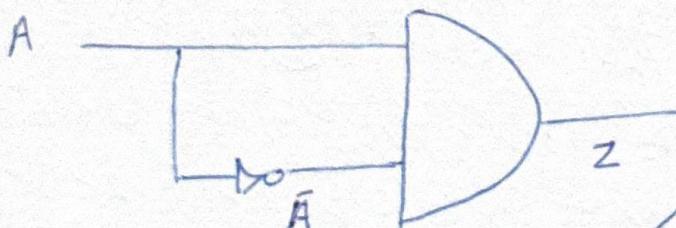
۵ هنگاهی زمانبندی بحث سُرایط ۲ و ۳ از موارد بالا صورت نمایند، می‌توانیم مدرج زمانبندی قبضه‌گار (عین اضافه‌گاری) است. در این نوع زمانبندی، فرآیند در حال اجرا می‌تواند توسط سیم‌عامل متوقف شود و به حالت آماده انتقال باید.

۶ سیم‌عامل‌های مدرن نظیر Windows، Linux و macOS از طرح‌های زمانبندی قبضه‌گار (عین اضافه‌گاری) است که در اینجا لفظ دارم به ذکر آسنا که طرح‌های قبضه‌گار در حالت کل سرور را بسیار راحت تجھیل می‌نمایند این حال خوبست همچنان را بروی کاربران ارائه می‌دهند.

با اینحال، زمانبندی قبضه‌گار دائم مسئله‌است که در ادامه پیش‌آمد اینها اینجا لیست شوند.

مسئلہ ۱: هنگاهی که چندین فرآیند، داده‌های را به استرکت گذاشتند، زمانبند قبضه‌گار سببی نوکر سُرایط روابط (Race condition) بین آنها اتفاق بیفتد آن‌چنان سُرایطی که درستی صدیرمت نشود و ترانزیت را در وضعيت ناطلوب قرار دهد و وضعيت خروجي واسمه به ترتیب اجرای فرآیند‌ها تغییر کند.

مثال ساده - سُرایط روابط: صادر زیر را در نظر بگیرید:



در این مدار، سیم‌عامل خودش and می‌شه و وضعيت ناطلوب خروجي از نظر ما تولید صدارت ~~صادر~~ $Z = \text{false}$ است. حال فرق کنید که صدارت A از false به true تغییر کند، درین وضعيت از ~~فرآیند~~ انتصار A به سبیللی Not تاخیر داشته باشد، خروجي $Z = \text{true}$ در وضعيت ناطلوبی را در خروجي رقمی زند.

مثال مرکبی - سرالید رقابت: در فرآیند مرداده از رابه استرآر نهاده است اند در نظر دلیل نیست. فراغت ملیند در حالکله که فرآیند در حال بروز رسانی را ده است، قبضه سده و غرآندر تیرا جراحتی شود. فرآیند قوم در حین اجرار خود دارد استرآر کی راهی خواهد داده بیش فی آید.

مسئل ۲: همچنانه ما بیندر قبضه از بر اوی مهاری کرزل ستم عامل نتراتاسیزی گذارد به عنوان مثال سرالید در نظر گیریم بر کمی فراخوان سیمه در عصاینه اجرار خود است. در این وصفیت قبضه کردن فرآیند در حال اجراء محلن است که داده های محظی کرزل را تغییر دارد و سبب ناسازگاری (هرچو و هرجو) در عملکرد ستم شود.

۹ فرآیند های کرزل ستم عامل نتراتاسیزی به دو صورت ^{که توانند} قبضه از عین قبضه زمانی شوند.

- زمانبند بیندر قبضه (اعضاواری) برای فرآیند های کرزل که سرالید مانند زیر صورت های ذیره:
۱ انتظار برای تکمیل سکن که فراخوان سیمه سیمه،
۲ انتظار برای تکمیل سکن که ۵/۱ در خواسته.

> طرح بیرون قبضه، ساختار کرزل از آنجا سیکم کرزل سبب قبضه کردن فرآیند ها ایجاد سرالید رقاقب (ناسازگاری) نمی شود، ساده تر است.

- زمانبند قبضه از (عین اعضاواری) برای فرآیند های کرزل، برای جلوگیری از سرالید رقاد (Race Condition) به قطام ریسنسی به ساختار های داره از استرآر کرزل از مکانیزم هایی مانند قفل های *mutex* استفاده می کنند. اگر ستم عامل های مولن برای زمانبند فرآیند های کرزل از مکانیزم زمانبند قبضه از استفاده می کنند.

۱۵) وقفرها و کدهای استرائی

از آنجلیکه و قفرهای توانند در هر زمانی درست رخ کا سوری رخ دهد و نی توان ممکن است این را ناریده و نفت، می باشد بخشی از کد از استفاده همزمان حین دین فرازند اجرای مخالف است مگر آنکه از حینی می قطعاً در سریانه فذ کو رخ ندهد ممکن است سریانه نامطلوبی صورت گیرد.

صورت زیر به لیار سود:

۱) داده ورودی از بین برود،

۲) داده ای روی خروجی در بازه نرسید سود و داده قبل از بین برود.

بنابراین بخشی هایی از کدنی توانند توسط حین دین فرازند به صورت هم و نداشفاده سود و از بین رو و قفرها را به هفتم ورود بین بخشی غیرفعال و به هفتم خروج فعال می کند. بر طور معمول بخشی هایی از کد در حینی هستند دارای دستورات کمی پاسخ دارند.

۱۶) توزیع لنده (Dispatcher):

توزیع لنده می دسراز صولفهای زمانبند (PLC) است و در واقع صادری است که لنده هسته پردازشی را در اختیار فرازند استخاب سده توسعه زمانبند (PL) می راند. از جمه وظایف کلیدی این صولف عبارتند از:

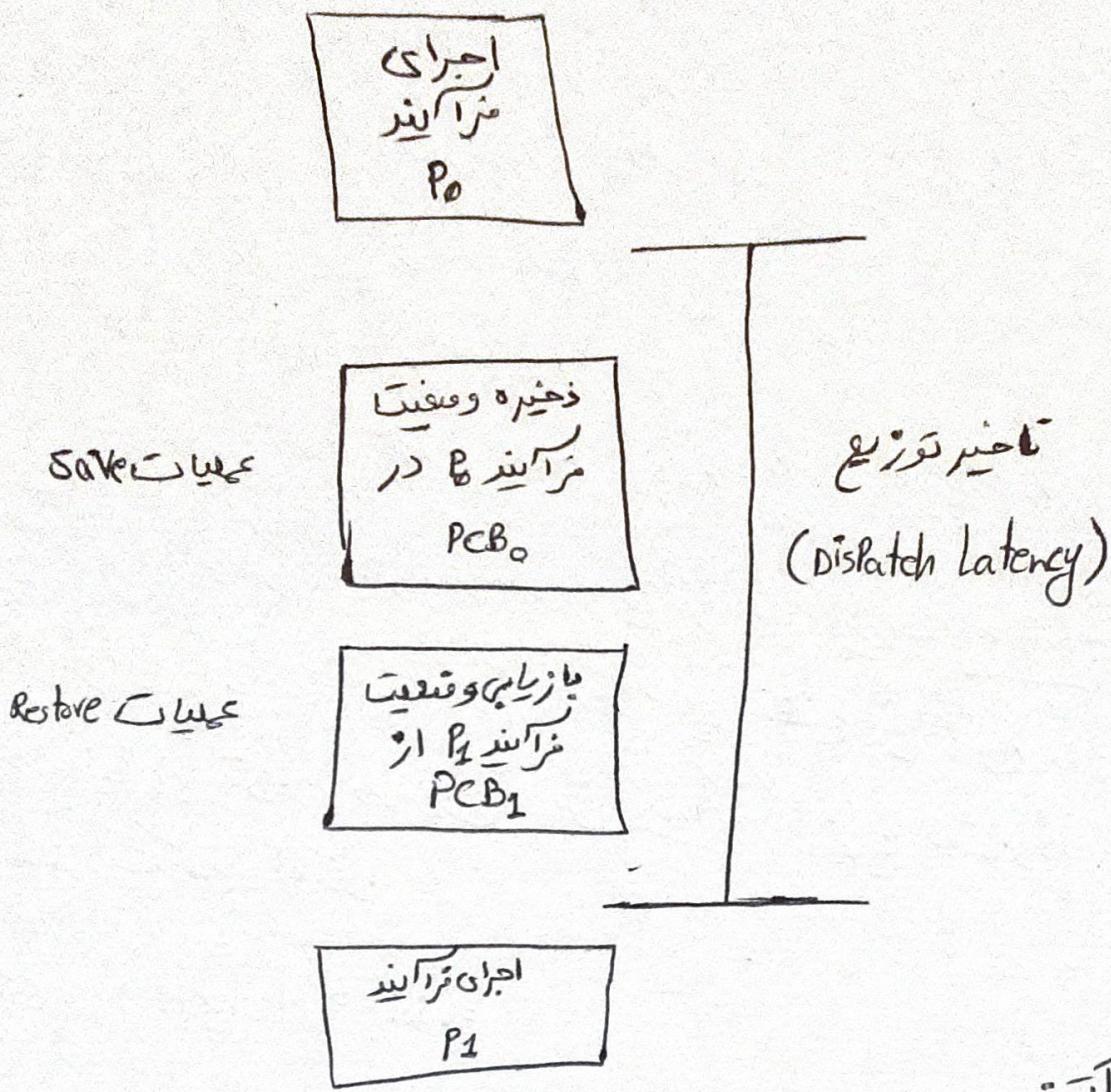
۱) سویچ کردن متواری فرازند با فرازند گیر (Content switching)

۲) سویچ کردن به مرکزی،

۳) پرس ب مقام مناسب از برنامه کامپیوتر برای اجرای بزایم.

تأخیر توزیع (Dispatch latency): پر زمان اگر فرآورده سده توسعه توزیع لنده پر زمان توقف یک فرازند و می سردی فرازند پرس، تأخیر توزیع اطلاعاتی می سود.

نکل زیر، تاخیر توزیع را به صورت مصور نشان جی (هد)



از آنجاییست، توزیع لغزده می باشد در هر تعریف متن فراخوانی شود، ۱۳

با در تابع کان سریع باشد

سوال: تعریف متن ها در حیث بازه زمانی درست کامپیوئری ۱۴

پاسخ: درست عامل های سینوسی می توان بازه تعریفی متن را در دوره زیر مسأله زمانی

گرد کرد در این مرد از آنها سریع جی شود

① در سطح کل سینوسیم

② بولی یک مرآیند خاص.

تعداد تقویف متن در سطح سیستم: نحوی رستور زیر، تعداد تقویف متن های در سیستم شامل
لینوکس نشانی دهد.

Vmstat 1 3

جایگزین:

Virtual Memory Statistics Reporter Vmstat —

است و از این میتوان برای این فرآیند که وظایف پردازشی و نمایش خلاصه اطلاعاتی
(در اینجا با نام آنها، حافظه، صفحه‌نیزی، I/O کاتله‌ها و خواسته CPU را بیان کنید) مفهوم

— عدد ۱ بیانگر تاخیر امکانی برای مسیمه تقویف متن های در حدود
سده در ۱ گانه اخیر است.

— عدد ۳ بیانگر نسائی خود در در حدود است که حداقل اطلاعات
مربوط به هر چنان تقویف متن های از زمان boot تا کنون
است و حدود دوم و سوم بر حسب پراویت دوم (زمان تاخیر)
به ترتیب تعداد تقویف متن های در ۱ گانه اول و در گانه دوم را
نشانی دهد.

جزءی حاصل از اجرای رستور بالا، نوع سیستم کنفرنسری موجود زیر جزو است:

۵۷۱

صیانتی تقویف متن از زمان boot →
سیستم عامل تا کنون

۱۴۹۴

→ تعداد تقویف متن های در ۱ گانه اخیر

۱۴۰۵

اخیر (بعد از اجرای رستور بالا)

تعداد تقویف متن های در ۱ گانه

نقیم بعد از اجرای رستور بالا

تعداد تقویف متن برای یک فرآیند خاص، برای مشاهده تعداد تقویف متن ها برای یک فرآیند مسکونی می باشد از دستور زیر درست عوامل های لینوکسی که بروز رفت:

cat /proc/pid/status

جایی که PID بیانگر نام فرآیند صورت نظر است و می توان با استفاده از دستور ps -el که در بخش های قبلی معرفی شد متن فرآیند صورت نظر نشانه آن را یافته.

ترجمی حاصل از اخراج دستور الاء شامل جزئیات سطراست که در میان آنها، در سطر زیر بیان مادا از اهمیت است:

Voluntary_ctxt_switches: 5692

non Voluntary_ctxt_switches: 1042

یک تقویف متن داوطلبانه هنگامی حارثی سود که فرآیند به سبب نیاز به فناوری هم آنقول در استرسن سنت استفاده از CPU را ترک می کند.

در مقابل یک تقویف متن غیر داوطلبانه هنگامی حارثی سود که CPU بنا بر این مدل انعقاد زمان اجرای فرآیند را لغو کند بالاتر فرآیند را بر اجرای فرآیند حارثی گرفته سود