

برای اجرای Thread نیاز به یک سیستم عامل داریم و در حقیقت سیستم عامل است که باید در دسترس باشد

این کار خود را در process انجام می‌دهد

TLB نیازی به چک کردن ندارد چون فضای حافظه کش شده

افزایش تعداد Thread های مختلف طبق random نیست

برای این که بتواند Thread ها را مدیریت کند که در OS این کار را می‌کند

اما با توجه به این که کار می‌کند

OS یک هسته مرکزی است و در داخل فرایند کاری ندارد و فرایندهای مختلف بدون آن نمی‌توانند

که از طریق System Call می‌تواند این کار را بکند

حقیقت این است که Thread ها مستقل از یکدیگر هستند و می‌توانند به یکدیگر دسترسی داشته باشند

Thread برای هر یک از محاسبات می‌تواند اطلاعات یکدیگر را در حافظه کش خود داشته باشد

طریق فضای مشترک است (که به ارتباط process ها از طریق message است)

## مزایای Thread

کمتر نیاز به پردازش دارد برای اجرای کارها

امتیازاتی که به هر یک از آنها می‌دهیم و می‌تواند به نفع آنها باشد

حقیقت این است که ارتباط بین آنها به Message خیلی سست است

اما هر یک از آنها می‌تواند به نفع دیگری باشد

مثلاً وجود caching، اطلاعات را به ازای کش می‌آوریم و در حافظه کش می‌گذاریم (سخت‌تر است)

در هر یک از این موارد ما یک اطلاعات را به Cache می‌آوریم و به نفع دیگری می‌تواند از آن

استفاده کنند پس به نفع دیگری می‌خورد پس می‌توانیم خیلی سریع به آنجا برویم



PC : Thread

Stack ✓

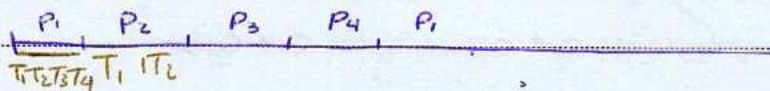
register ✓

spread sheet ✓

دکتر بهمنیار حاج احمد نوری

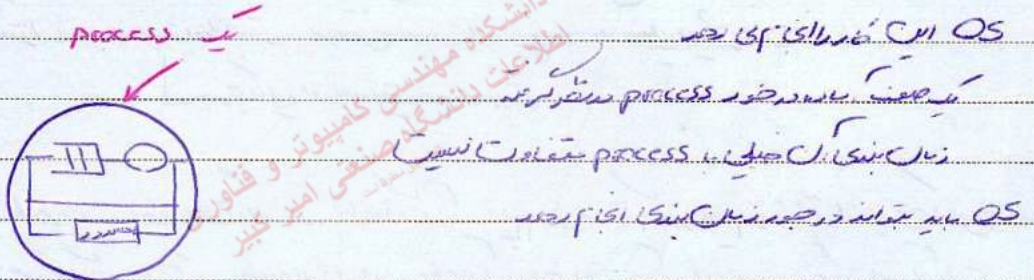
(مجله ۴ کتاب Stalling, Silberschatz)

نویسنده: P<sub>1</sub> - P<sub>4</sub>



Thread P<sub>1</sub> در سیستم عامل Thread در سیستم عامل، این زمان به این Thread تقسیم می شود. در سیستم عامل، این زمان به این Thread تقسیم می شود. در سیستم عامل، این زمان به این Thread تقسیم می شود.

OS در سیستم عامل، این زمان به این Thread تقسیم می شود. در سیستم عامل، این زمان به این Thread تقسیم می شود. در سیستم عامل، این زمان به این Thread تقسیم می شود.





1 point  
برنامه‌ریزی سیستم‌های حریف state machine به کار می‌آید (بعد از اجرای مجموعه دستورالعمل تعیین شده است)  
دری‌نجه‌ها برای این دستورالعمل را اجرا می‌کند.

غالباً کاربر می‌کند بیشتر CPU ای‌ام بعد توسط سیستم عامل ای‌ام می‌شود مثلاً توقف برنامه

از آن \* درگاه به برنامه‌ریزی را در اختیار دارد چه در این سیستم می‌تواند به برنامه‌ریزی و برای سیستم‌های اصلی این

زمان اصلی طولی می‌شود \* عملیاتی است \* floating point \* spelling check  
عملیاتی طولی \* عملیاتی طولی

نمایش \* اینده یک برنامه در سیستم CPU را در اختیار دارد چه در این سیستم می‌تواند به برنامه‌ریزی و برای سیستم‌های اصلی این

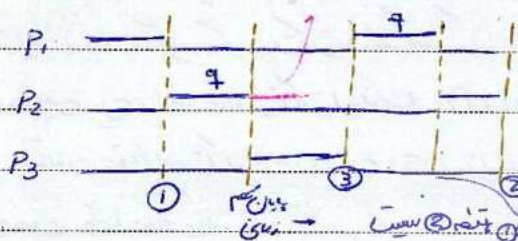
مثلاً برای برنامه‌ریزی یک web page صبر می‌کند تا کلیک می‌شود اتفاقی می‌افتد به این علت است که عملیاتی است  
CPU در اختیار برنامه‌ریزی می‌شود \* در این سیستم طولی می‌شود

به این احتمال دارد که برنامه‌ریزی از این سیستم به کار می‌رود \* در این سیستم طولی می‌شود (interactive) می‌شود در این

در این سیستم interactive می‌شود در این سیستم طولی می‌شود \* در این سیستم طولی می‌شود

تفاوت اصلی این سیستم Time sharing است؛ یعنی هر چه در خواست می‌کند برای برنامه‌ریزی را در اختیار می‌گیرد و در این سیستم  
مجموعه‌ای به تعداد سیستم‌ها که می‌تواند اجرا شود به کار می‌رود و به کار می‌رود (مجموعه‌ای به تعداد سیستم‌ها که می‌تواند اجرا شود)

Time sharing \* در این سیستم طولی می‌شود \* در این سیستم طولی می‌شود



کمتر از 9 (در این سیستم طولی می‌شود) \* در این سیستم طولی می‌شود

این در این سیستم طولی می‌شود \* در این سیستم طولی می‌شود

در Time sharing می‌تواند برنامه‌ریزی را در اختیار می‌گیرد و به کار می‌رود (مجموعه‌ای به تعداد سیستم‌ها که می‌تواند اجرا شود)

در این سیستم طولی می‌شود \* در این سیستم طولی می‌شود

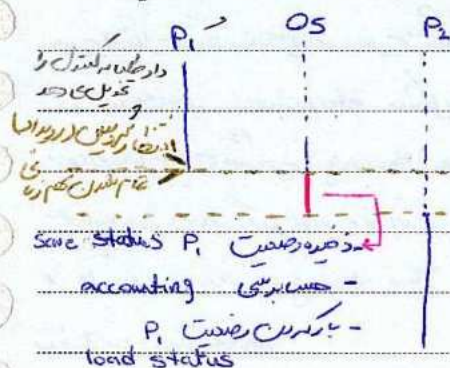
1 point \* در این سیستم طولی می‌شود \* در این سیستم طولی می‌شود



$P_1$  \_\_\_\_\_  
 $P_2$  \_\_\_\_\_

مستقیم عمل میاید بترانه چندین باره صحت میسرند اگر اندکی در عمل بیشتر دارند. قطعاً یک عمل را در هر کف می آورند  
اجزای دلی در کف نظیر ریاضی است. چگونه عمل است یک باره بیشتر میاید را در وقت طولای در اختیار داشته  
باشد و کف بر سه نیمه باشد و این صاف قطعاً کف برای سطح است

المشقة



عبدالرزاق ابن ابی نعیم

$P_1$  کا وزن  $P_2$  سے زیادہ ہے

کتابخانه مرکزی و اسناد خطی و مطبوعاتی

بعد اکتشاف  $P$  ملزم به  $P_2$  قبول به حد خود را بخاطر

این کار را می‌تواند OS این کار را می‌تواند OS در حال حاضر

۲. گفتار را بدین سیاقی بنویسید که کوتاه، طاری باشد و گفتار را به  $P_2$

جی ریڈیو، الٹرا وائٹ، انفر ریڈ، وائٹ، گری، بلیک، اور پورے طیف میں  $P_2$  وائٹ کنٹریل ہے۔

۲. یک اصل از برای است که در این کتاب OS نیز بیان شده باشد و اصل را در

• int switch رانش

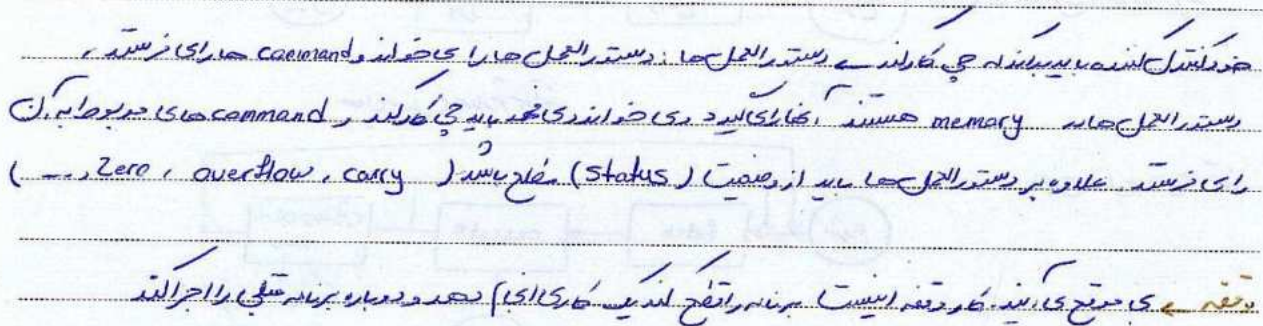
OS به هر گزین وضعیت  $P_2$  کنترل را به  $P_2$  والداری نهد پس OS این کار را ایفا می کند چون  $P_2$  و  $P_3$

عفی عنہ فی السنۃ، راجعاً

OS تا پروازنده را در اوضاعی تغییر دهد هیچ کاری از آن ساخته نیست پس OS همیشه تا ابد در اجزای سیستم یا در حال

احساسیت





microprogram (البرنامج الميكرو)

۱۔ بہ خاطر سرعت یا سبب کی وجہ سے رفتہ

→ پہلے مشہورہ (observable) ہے جبکہ اگر نہ تو اس کی توقع نہ کی جاسکتی ہے نہ اس کی تصدیق کی جاسکتی ہے۔

Stack pointer, address reg, Data reg

لایروتن نشینده (nanobscorable) - به اندازه‌ای که از این‌ها می‌توان استفاده کرد، زیرا زیست‌پایدار نیستند.

خط و سبک های نوین از این جا استفاده کنید و به تبدیل ها کا حاتم و وضعت

در و پیش از رسیدن از حدیثی که در کتاب است

که به سستی در بدن ۵۵ باشد می توان از دانه های زیر استفاده کرد

۴۹. صدای ریش آشفته و سستیم حاصل از این ها کجری اند (رضیت از صدای انداختن یا کشیدن)



Subject:

Year. Month. Date. ( )

چگونه عملیات این reg ها برای OS رید و وایت برای P, D, P قابل مشاهده نیست از بی نظیر کدام برنامه است که اجازه بدهد و چه کارایی پیدا کند؟

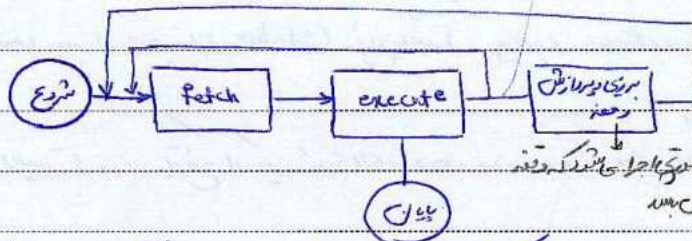
نحوه اجرای دستورالعمل در فستورال

سیکل Fetch و اجرا: دستورات از memory به حافظه کشیده می شود و سپس ارسال command به واحد کنترل کننده توسط واحد کنترل کننده. سیکل است که طایفه اجرای دستور (دستورالعمل جدید، دستورالعمل بعدی) ...



داخل بودن پردازش و قفله

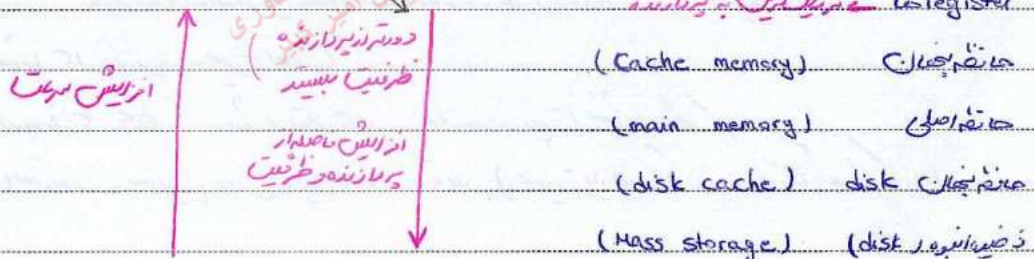
ماژول داخل بودن و قفله



در پایان اجرای دستورالعمل یک بررسی می شود که به قفله می افتد یا نه. به این چیزی که می بینیم می بینیم که اگر نه به بیرون می رود و دستور بعدی می بینیم پردازش و قفله. بعد از اجرای دستورالعمل در وسط سیکل Fetch اجرا می شود. پردازش و قفله های خارجی در پایان اجرای دستورالعمل (دفعه ای داخلی هم داریم) اگر و قفله ها نباشند امکان پیاده سازی کمتری وجود ندارد.

حافظه و حافظه سازی

حافظه در سیستم طبق سلسله مراتبی است (hierarchical) حافظه های cache در این خاصیت می باشد





حافظه‌های سلسله‌مراتبی ← علاوه بر حافظه درون فرآیند سرعت هم بالا برد و دسترسی سیستم را به حافظه بالا ببریم  
(در register ها) ← حافظه‌های disk که می‌کنیم بیشتر به reg ها دسترسی کنیم

← حافظه‌های ذخیره‌سازی را چه گویی می‌کنند؟

register ها: compiler (دیتای‌های سطح بالا)، خود برنامه‌نویس (دیتای‌های سطح پایین)

حافظه اصلی: سیستم عامل (دیتای اصلی که سیستم می‌خواهد) (حافظه اصلی)

disk: سیستم عامل ← در سطح رابط block ها تقسیم بندی می‌کند برای یک file directory می‌سازد

حافظه‌های cache: خودشان، خودشان را کنترل می‌کنند چون یک حافظه کنترل کننده همراه خودشان دارد در OS

اصلی از خود چنین چیزی ذخیره ندارد گویی نه cache را می‌بیند و کنترل می‌کند خودشان در کارها در حافظه می‌کنند و کارها را

سرعت ببرند ← کنترلش سخت‌تر است

حافظه‌های disk: کنترلش با خودشان است و نه می‌کنند گویی با خود به disk دسترسی پیدا کنند و کارها را با حافظه

کریل سرعت می‌گیرند و می‌توانند

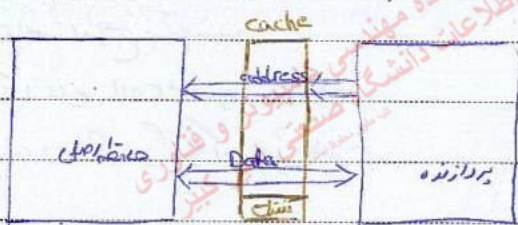
← OS حافظه اصلی با disk را می‌بیند از خود حافظه‌های اصلی خبر ندارد

حافظه اصلی

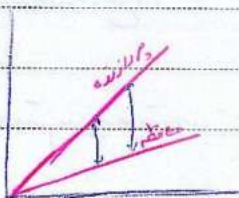
سیستم عامل برای کار خود نیاز به داده دارد و این داده‌ها را از حافظه اصلی می‌گیرد

حافظه‌های cache

فرآیند می‌توانیم سرعت بالا برد حافظه سلسله‌مراتبی به این دلیل که هر چه می‌خواهیم در حافظه‌های اصلی پیدا کنیم



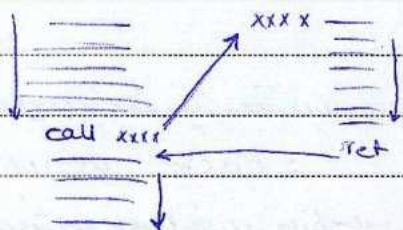
حافظه‌های اصلی و ثانویه در حافظه‌های اصلی و ثانویه





پرونده در اصل در محل چهارم به خط اصلی می‌رود. در محل چهارم آن فرستاده می‌شود. cache در این  
 وسط قرار گرفته. در محل پنجم به خط اصلی می‌رود. در محل ششم به خط اصلی می‌رود. در محل هفتم به خط اصلی می‌رود.  
 می‌فرستد. بخش‌های از خط در cache می‌شوند. در خط پنجم به خط اصلی می‌رود. static استفاده می‌کند  
 و سرعت بالا را می‌دهد. در خط ششم به خط اصلی می‌رود. در خط هفتم به خط اصلی می‌رود.  
 اصل کلی (locality) هم از نظر زمانی و هم از نظر مکانی می‌باشد.  
 اولی که می‌گوید به خط اصلی می‌رود. در خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود.  
 که می‌گوید در خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود. در خط هفتم به خط اصلی می‌رود.  
 بخش‌های این است که از خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود. در خط هفتم به خط اصلی می‌رود.  
 دسترسی داریم و میزان موفقیت ۹۵٪ است.

cache را می‌توانیم به خط اصلی می‌رود. در خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود.  
 این می‌گوید که در خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود. در خط هفتم به خط اصلی می‌رود.  
 چیزی را از خط پنجم به خط اصلی می‌رود. در خط ششم به خط اصلی می‌رود. در خط هفتم به خط اصلی می‌رود.



return و call

این می‌گوید



در حال حاضر هم خط پنجم است و هیچ چیزی ندارد.  
 فرآیند call به خط پنجم می‌رود و خط ششم به خط پنجم می‌رود.  
 در خط ششم به خط پنجم می‌رود و خط هفتم به خط پنجم می‌رود.  
 در خط هفتم به خط پنجم می‌رود و خط ششم به خط پنجم می‌رود.  
 در خط ششم به خط پنجم می‌رود و خط هفتم به خط پنجم می‌رود.

این می‌گوید



← قطعه OS بر اساس وقفه کار می کند (به سبب قوی است)

### دسته بندی وقفه ها (interrupt & exception)

1. اجرای برنامه: سلاهای زمان در حال اجرا است و عملی ای می باشد که می تواند چه کار کند (تقسیم حافظه، استفاده از دستورالعمل تعیین شده، مدیریت حافظه، overflow، ادغام خروجی از فضای تحریف شده، پردازش داده ها و بی جا کردن برای اینجا باید چه کار کند؟) تمام دفعات پردازش در این منابع تولید وقفه است و تولید وقفه باعث قطع برنامه می شود و به جای پیگیری ورود (توقف تولید وقفه می کند و کار دیگری می کند این بدترین وقفه است) که ظهور این امر می دهد. با تشخیص می کند به برنامه را قطع می کند (به پردازش عمیق می رسد برای این کار پردازش و یک برنامه کار از این ظاهر را می گیرد) و در صورت اجرای دستورالعمل قطع می شود

← عکس العمل پردازنده در برابر خطا یا عملیات غیرکارساز (تولید وقفه)

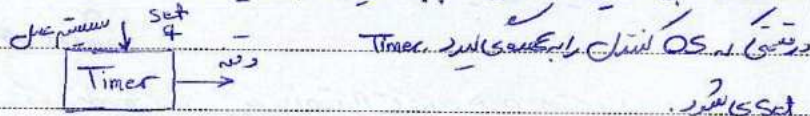
2. زمان اجرا (Timer): هر برنامه یک کم زنی دارد و یک Timer وجود دارد که وقتی از زمان اجرای برنامه تمام شود تولید

وقفه می کند و به سیستم ارسال می کند و همان برنامه نیست

← کم زنی را می توان در (تنظیم از کار کرد عملی) که آن را روی زمان توقفی Set می کنیم تا زمانه

کم زنی را (19) شروع به کار می کند و هر وقت به زمان 4 رسید تولید وقفه می کند OS این را

Set می کند و قبل از این که گذشتن را به برنامه پیگیری بعد از Set می کند



3. ورودی و خروجی (I/O): هر سیستم کامپیوتری یک یا چند port دارد که IO

اتصال به فضای خارج از طریق این port ها است که برای خود وقفه دارند

برنامه ای منتظر اطلاعات ورودی است از یک منبع که اطلاعات دارد و منتظر است که گذشتن را

در محله OS است. در در اطلاعات خود را باید خود را به پیگیری کند و در در اطلاعات



وقفه می دهد و به پیگیری می رسد و در در حال اجرا است که این را در در اطلاعات خود می کند و اطلاعات ورودی را در در

از صفات پذیرنده و آن کار برای برنامه منتظر است اما بعد از گذشتن وقفه های IO پایان عملیات ورودی و خروجی را اعلام

می کنند و پایان خروجی ← اطلاعات کار خارج شده و در در اطلاعات می رود که به سیستم



4. خرم رفته به دستم بجای ری ندارد (دریاد) و این سینه می ۵ مربوط بودند

4. فرض محال (H.W. Parity) می‌تواند طوری ساخته بشود که در صورت قرار گرفتن تولید و قفله‌ها در یک مسیر، راه برای حلقه وجود ندارد. Parity که در صورت استفاده تولید و قفله‌ها در یک مسیر، راه برای حلقه وجود ندارد. فرض محال (H.W. Parity) می‌تواند طوری ساخته بشود که در صورت قرار گرفتن تولید و قفله‌ها در یک مسیر، راه برای حلقه وجود ندارد. فرض محال (H.W. Parity) می‌تواند طوری ساخته بشود که در صورت قرار گرفتن تولید و قفله‌ها در یک مسیر، راه برای حلقه وجود ندارد.

وقته نوع اول در صدد اجرای دستور العمل می تواند به نام برآید و طرح اند چونکه دستور العمل غیر قابل است و در اصل می تواند  
از راه های آمد و رفت و در صدد اجرای دستور العمل تا آخر اجرای دستور

در منع اول وقف باید دوباره بطلان التخل فلیکبر حال اجرای استثنای استثنای در سلبه وقف حاصل و بطلان التخل حاصل  
اجرای استثنای تخریب شود به استثناء اجرای در سلبه استثنای حاصل و وقف با اجرای التذ

برنامه‌های کاربردی نمی‌توانند با port ها کار کنند و باید از سیستم فایل جداگانه‌ای (فایل‌های جداگانه) استفاده کنند. در این سیستم فایل، هر برنامه‌ای که می‌خواهد با port ها کار کند، باید از سیستم فایل جداگانه‌ای استفاده کند. در این سیستم فایل، هر برنامه‌ای که می‌خواهد با port ها کار کند، باید از سیستم فایل جداگانه‌ای استفاده کند.

(Dual mode) میں دو طرح کی

چگونه می‌توانیم دو mode اجرای فایل را به ما اختیار کند؟ می‌توانیم به عنوان یک خط اول در فایل خود به این صورت بنویسیم:

کتاب احتیاجات زیاده راجحی بسیاری دارد و در هر یک از این موارد می تواند کمک کند

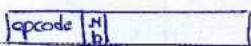
System mode : (اختیارات زیاد دارد) (مستثنی) Set, port, Timer (۲ - ۳)

user mode : اختیارات محدود



با `port` های ورودی خارجی از طریق دستگاه های ورودی خارجی می توانیم کار کنیم. اندازه `port` ها 16 بیت است.  
در هر زمان که می خواهیم با `port` کار کنیم باید سیستم عامل بخواهد.  
در سیستم های تران با `port` ها `reg` های که می توانیم کار کنیم در همه سیستم های تران با `port` ها کار می کنیم.  
در همه سیستم های تران با `port` ها کار می کنیم و به `port` های که می توانیم کار کنیم در همه سیستم های تران با `port` ها کار می کنیم.

بیت در دسترهای اول  $a_{code}$  اضافه می شود.  $made\_bit$  نشان دهد  
در جای دسترهای اول  $a_{code}$  است.  $a_{code}$  در جای دوم قرار  
دسترهای آخری  $a_{code}$  در جای دوم قرار می گیرد.



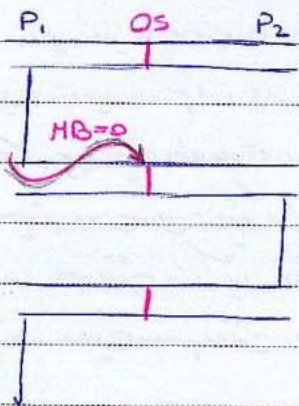
دفعه ۱ بار MB باید صفر باشد (در reset و اختیارات سیستم و زمان در ۰.۹۰۰ بیکی مانند جعل شروع خط ۰.۵ است)

حالاتی خواص انتقال را به برنامه ای میسر می دهد قبل از اینکه برنامه

توین بعد از آن می تواند عوض کردن Mode bit از دستورات OS

است و وقتی در کنترل دوباره ای افتد دست OS خودش صفر می شود

بعد از آن خواص Mode bit را می تواند برنامه ای شود



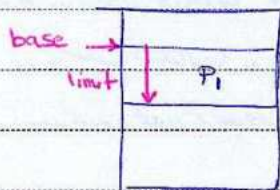
دانشگاه مفهومی کامپیوتر و فناوری  
ملاعات دانشگاه صنعتی امیر کبیر



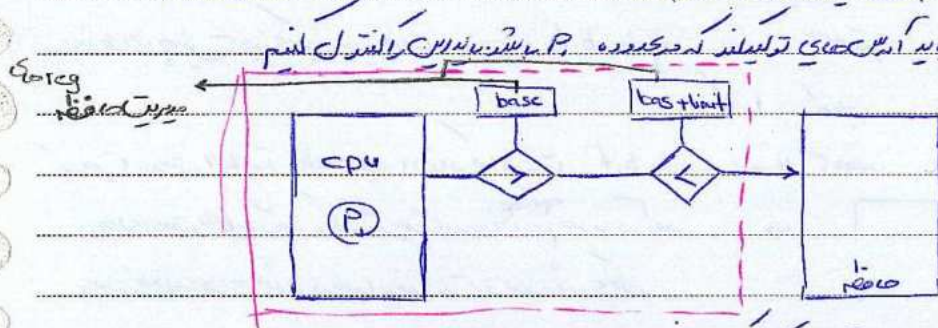
Subject:

Year. Month. Date. ( )

بسیاری از پردازنده‌ها چون به سخت تغییر data می‌دهند (بسیاری از پردازنده‌ها به سختی می‌توانند داده‌ها را تغییر دهند) و به همین دلیل به سختی می‌توانند داده‌ها را تغییر دهند.



هر برنامه یک آدرس base دارد که محل شروع برنامه و یک limit دارد که طول برنامه است. هر برنامه باید این دو عدد را اختیار کند و OS نسبت به این دو عدد (محدود می‌کند) که هر برنامه می‌تواند از این آدرس استفاده کند و هر برنامه می‌تواند از این آدرس استفاده کند. در این حالت می‌تواند به آدرس‌های دیگر دسترسی داشته باشد.



عمل از این جهت است که به صورت یک شرط را چک کنیم تا بفهمیم که می‌توانیم به آن آدرس دسترسی داشته باشیم یا نه.

1. باید کمتر از آدرس base باشد.

2. باید بیشتر از آدرس  $base + limit$  باشد تا در محدوده  $P_1$  باشد.

این مقایسه باید به سختی از طرف این سیستم انجام شود. به عنوان مثال در این سیستم مقایسه می‌شود که آیا آدرس در محدوده  $base$  و  $limit$  است یا نه. این مقایسه در این سیستم به سختی انجام می‌دهد و به سختی می‌تواند به سختی مقایسه کند. به عنوان مثال در این سیستم مقایسه می‌شود که آیا آدرس در محدوده  $base$  و  $limit$  است یا نه.

سیستم عمل از این جهت است که به صورت یک شرط را چک کنیم تا بفهمیم که می‌توانیم به آن آدرس دسترسی داشته باشیم یا نه. این مقایسه در این سیستم به سختی انجام می‌دهد و به سختی می‌تواند به سختی مقایسه کند. به عنوان مثال در این سیستم مقایسه می‌شود که آیا آدرس در محدوده  $base$  و  $limit$  است یا نه.

در هر برنامه برای این حالت نقش دارد که به این reg (دیده نشود) هستند.

درست العمل می‌تواند به سختی از این reg (privilege) هستند.

privilege ← درست العمل می‌تواند به سختی از این reg (privilege) هستند. Mode bit رتبه‌بندی می‌کند. هم یک درست العمل privilege است.



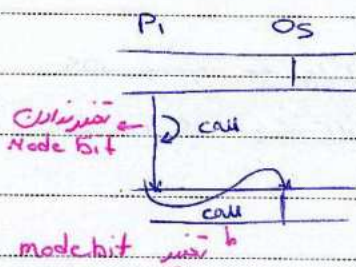
به طور کلی برنامه‌ها فقط دستورالعمل‌های تعیین شده اجرا می‌کنند و می‌توان آن را طوری طراحی کرد که در مقابل کارهایی که به نیازهای آن در هم تداخل دارند پس از آن به هر دو در وقت خود رسید.  
یعنی جوابگوی آن در شرایط عادی به صورت تولید وقت است.  
سیستم عامل در صورتی می‌تواند خود کار کند که دارای سیستم قوی وقت باشد.

در حالتی که  $P_i$  کنترل اجرا از دست داده و به OS برسد.  
در وقت انتقال به خود می‌تواند برنامه‌ها را اجرا کند و می‌تواند یک چیزی از OS بخرد و در وقت خود به آن برسد.  
از سیستم عامل می‌تواند تا حدی که می‌خواهد استفاده کند و می‌تواند از سیستم عامل استفاده کند.  
این وقت که می‌تواند وقت هم برای آن است.

وقتی که در زمان اجرای برنامه از OS بخواهد استفاده کند تا بتواند به برنامه برسد و در وقت خود به آن برسد و اجرا شود و به آن جواب در وقت خود کنترل بر آن برنامه منتقل می‌شود.

(point) برنامه‌های وقت جز OS هستند و در وقت خود OS تعیین شده‌اند.

چگونه می‌تواند در وقت خود استفاده کند؟ احتمالاً  $\text{call}$  یا  $\text{routin}$  می‌کند.  
این  $\text{call}$  یا  $\text{routin}$   $\text{call}$  همان وقت دارد چرا که سایر  $\text{call}$  ها که در وقت خود به آن هستند  $\text{Mode bit}$  را عوض می‌کنند و  $\text{call}$  برای در وقت خود به  $\text{Mode bit}$  را عوض می‌کند.



برای  $\text{call}$  ها  $\text{system call}$  می‌کنند.  
 $\text{System call}$  به معنی  $\text{call}$  است که برای OS هستند و از طریق آن می‌توانند به OS برسد.  
مثلاً برای اجرای برنامه‌ها می‌توانند به OS برسد.

نوع  $\text{call}$  می‌تواند  $\text{system call}$  یا به طوری باشد که می‌تواند از آنرا استفاده کرد و  $\text{System call}$  به  $\text{Mode bit}$  را تغییر دهد.



نکته ۱. Intel برای این طریق دستورالعمل تحویل کرده که نام آنرا Interrupt گفته و آنرا با عدد صدای کند

INT XX (58)

Intel یک جدول INT را دارد به طوری که 58 (صدای 58) صدای INT (است) می دهد و در سطح کاربری دارد

و در این برنامه را از این اجرای کند پس همان call چیست که این call به در سطح غیر سیستم است

(چه مثال دفعه اولی کاره در اختیار سیستم عامل هستند)

حقیقی این اجرای شود به طور اتوماتیک Mode bit صفری شود و می بیند که انتقال را به برنامه برای اجرا می دهد به OS

ی بعد

درخواست از OS ← System call ← تغییر Mode bit به خود اتوماتیک (در برنامه برای دستور این تغییر

می دهد و برنامه دستور اجرای تغییر Mode bit نیست)

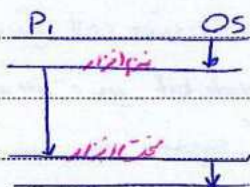
تغییر از حالت کاربر به سیستم قه از طریق Interrupt (دفعه ای می شود) (عبارت جمله ای شد D:)

در واقع دو انتقال است P انتقال را از دست داده می گوئیم (با دفعه)

تغییر Mode bit با رفتن از کاربر به سیستم از طریق کنت از دست

Interrupt به تغییر Mode bit می شود

را که اتوماتیک به صورت کنت از دست می آید این کارها



OS ← برنامه ← نرم افزار

برنامه ← OS ← کنت از دست



بمردانند و از این جهت در سوره النحل تحت این سوره ای آمده است

در برنامه ای که برای اجرای آن حدود یک میلیارد ریال را در اختیار میگیرد و با در اختیار گرفتن آنست که باید برای اجرای آن بود.

لغو از اتصال سیستم از جدید OS کافی هستند وکت الزامی انتقال می شوند پس صافه مکان یا cache

صدیق زائد

process management

1..  
реш. Сурж.

memory management

۱- صحت مندی

## Secondary memory management

میدان وادی خرمی

## Input-Output management

میری نئی

## File management

5. निम्नलिखित

Networking کنه آزاری ازین بین سیستم های مختلف

10) Синтез

## command Interpreter System

عربی از ۵۵ لغات به ۵۵ لغات دیگر

Рисунки

[illegible]

نمایندگرمینه در حال احیاست. یک مریضین داخل است. در ایندیوی اچ ای تی جی دارد.

process ← program → process  
البرنامج ← البرنامج ← البرنامج

اجراءات و پروسس process یک سوئیچ و پروسس دیگری است که یک سوئیچ و پروسس دیگری است

program آید و نسبت دکلن رست بکسل هم چو یکانه

در بین نرم افزارهای رایج برای سیستم های عامل OS اجزای اصلی درونی OS در دو بخش، بخش های کاربردی و بخش های مدیریتی تقسیم می شود.

نام خانوادگی: ...

صدیقہ از ایڈریس کٹرہا سٹیل میسور

(۱) ایاد حضرت زیند کابر سے فرمائی ہوئی صحیفہ داری

2- خلق کردن و پدید آمدن کردن ← خلق کردن موقفاً از حد و خارج این است. دنیا را در اختیار این نهادیم و به عین خود دنیا را

resumption      suspension



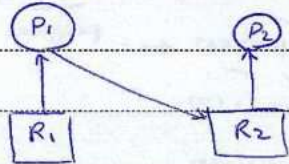
synchronization

۴. لطیف جوی برای آلودگی فرایند: بهانه ای سوال اجرا جندی را میسبب کرده این مقدار به درجه های دیگر مورد استفاده قرار گیرد پس باید در نظر داشته باشید که هر چه آلودگی زیان عمیق است دلی الزامی یک جبهه ای است و به بهانه ای که برای اجتناب دارد یا خود طایفه این بهانه هر وقت دلیلی خلاص است می تواند در آن استفاده کند در کامپیوتر تعجب در بهانه به کار بستن تمام در ارتقاء نیستند بلکه توسط ۵۰۰ هم در ارتقاء

## Dead lock

blocked

Swearing

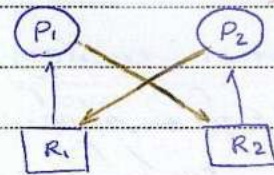


در این حالت  $P$  می تواند حالت گذر  $R_2$  را خلاصه کند  
 $R_2$  در اینجا پس از  $P$  می تواند به  $R_2$  برگردد  
 به  $R_2$  برگردد پس از  $P$  می تواند به  $R_2$  برگردد

این بنده خوار از دست او که در این لحظه که می نویسد که بجایه صبر ندارد و قفسه های اینده را می بیند  
سبحان اصبیح باری

blocked

blocked



یعنی  $P_0$  و  $R_1$  احتیاج پیدا نمی کنند، یعنی آن‌ها در آن کانتینر هستند چون  $R_1$  را گرفته،  $P_1$  است و چون در صف اجرای  $P$  است،  $R_1$  را می‌دهد.  
یعنی در حالت Dead lock می‌ماند.



این که OS مخد این حالت ایجاد افتاده کار واقعی نسبت به این که می شود محاسبه است راه حل ارائه شده است  
 کسی که این کار را می تواند راه حل ارائه دهد OS است چون چند  $P_1$  و  $P_2$  خواب هستند و غیر فعال است  
 می تواند برای این کار دهد

ویژگی خاص: **linear array**

حالت به صورت فیزیکی یک آرایه خطی از قطعات است  
 از آنجا که باید از ساختار **array** استفاده کرد به از هر یک از قطعاتی که توان استفاده دارد

ویژگی دیگر: **محدودیت**

1. یکی از ویژگی های استفاده شده در صاحب آن: یعنی باید به اندازه کم قطعات از حالت استفاده شده در صاحب آن می  
 است (خردی این کار را می تواند)

2. تخصیص حالت در سطح لایه آن

3. یکی از ویژگی های آن آزاد به بیان دیگر خاصیتی است که خطی است به این صورت به کار می رود

ویژگی خاص: **Disk**

خطی به صورت فیزیکی در نظر است

1. ویژگی تخصیص آن

2. تخصیص مکان ذخیره سازی

3. زمان بندی (زمان بندی) در سطح 2 تخصیص های در سطح 1 و 2 به صورت فیزیکی در سطح 1 و 2 در نظر است

در سطح 1 این را در این حالت می بینیم حالت به صورت فیزیکی در سطح 1 و 2 در نظر است

**scheduling**

در سطح 1 و 2 به صورت فیزیکی در سطح 1 و 2 در نظر است

مکان ها به صورت فیزیکی در سطح 1 و 2 در نظر است

در سطح 1 و 2 به صورت فیزیکی در سطح 1 و 2 در نظر است





Subject:

Year. Month. Date. ( )

حالت 1: زمان دسترسی مکان های بسیار است

حالت 2: زمان دسترسی مکان های تعداد است.

برای اینکه درک بهتر از زمان دسترسی را انجام دهیم از ترتیب بندی استفاده می کنیم مثلاً اول به ترتیبی به بخش می رویم که مکان آن به صورت نزدیک تر است

در این مورد بررسی خواهیم کرد:

یکی از کارهای مهم IO management مدیریت منابع است

1- مدیریت صف ها: برای اینکه بتواند ترافیک ها را از دست

2- کنترل ورودی ها و خروجی ها: مثلاً می خواهیم با port ورودی ارتباط برقرار کنیم قبل از این لازم است port را چک کرد و اگر می خواهیم خود را به port وصل کنیم باید چیزی را در مورد port بدانیم یکی از قبل کنترل اینها را برای جلوگیری

3- سیستم بار: بار زیادی به صف cache است اگر می خواهیم با port ها کار کنیم با بار زیادی داریم چون port ها انداخته و وسط این port و به بار نهاده هستند یکی به صف cache ها کنترل کنان افزای انداز پس که اصلی بار ترافیک بخشیدن به کار است. اداره یافتن اینها به کمک OS است.

در این بخش

نمای 1: دید کلی از اطلاعات را از این بخش

اطلاعاتی که می توانیم جاهای مختلف ذخیره کنیم (دری در سی دی) به نام Data می نامند؟ نه، اینها نیستند بلکه چیزی که می نامیم این اطلاعات هستند و یکی از چیزی که می نامند چون هر کدام از اینها چیزی از اطلاعات و به صورت اطلاعات چیزهای کلی اطلاعاتی که در این کار می اندازد و اینها را می نامند و اینها را می نامند



کلاس درس مدیریت سیستم

ایا در صورت

ایا در صورت راجع (directory)

که اینها برای کار کردن

جلسه سیستم : 7,9

## Command Interpreter System

سیستم مدیریت فرمان ها

و اما بخشی از OS نیست چیزی است که به آن سرور داده

در سطح ای برای کاربر به وسیله آن بتواند فرمان دهد

سیستم عامل مجموعه ای از نرم افزار است که به منظور

ایجاد آن تماس گرفتن با نرم افزار از سطح تقاضا

در سطح درخواست های ترانس مستقیم کار کنیم

محیط سیستم عامل درخواست های سیستمی است

پایان به فرین یک فرایند

exit ( )

که به OS می دهد این فرایند ها که تمام فرایندها را از اول

آغاز به فرایند کردن یک فرایند ( ) create

که منابع مورد نیاز فرایند را به سیستم می دهد

malloc ( )

سیستم مدیریت فرمان ها

به عنوان یک نرم افزار برای کاربر که به وسیله آن می تواند به OS دستور بدهد

نمیست در برای ای آن از تعداد زیادی System Call استفاده می کند

برنامه ها است که سیستم مدیریت فرمان ها است و در واقع سیستم مدیریت فرمان ها OS نیست و در فرمان OS می تواند

به کار خود ادامه می دهد



Subject:

Year. Month. Date. ( )

copy  $F_1, F_2$

خصوصی ایف بی

2

در حالت نام

$openFile()$  سیستم لایه است

بازگشت فایل ورودی

ایجاد فایل خروجی (به قاعده  $F_2$  تبدیل و جدولی داشته) → در صورت وجود فایل خروجی

→ در صورت نبود  $copy$  اطلاعات را از  $F_1$  به  $F_2$  می بینیم چون اطلاعات را نمی شناسیم در  $F_1$  از دست دادن

از بازرسیت داده ای کنیم و اندازه بازگردد در سیستم باید به جدولت به اندازه بازگردد در سیستم

حالت: از فایل ورودی بخوان

در فایل خروجی بنویس

سیستم خروجی

نمایش پیام

در فایل

→ در این جا error جا را مشخص نمی کنیم و خطا را می بینیم در صورت وجود خطا

باید برای سیستم خطا به خطا می بینیم

سیستم معین فرمات ها در واقع برای راحتی کار در سیستم OS هست

## System Calls

راخوان های سیستمی به جدول اینترپت های نرم افزاری هستند و این راخوان ها باید به جدولی compile شود و در همان Interrupt های نرم افزاری است و اگر قرار باشد این راخوان ها compile شود باید به این جدولی

تبدیل می شود

LD  $R_1, Parm$

LD  $P_2, Parm$

INT 28

$openFile()$

یک درون ارسال به اینترپت

استفاده از reg 19 در دست

روشن می شود ارسال به اینترپت از Stack است (push on stack)

این Interrupt های نرم افزاری می تواند به عنوان library routine و در جدول compiler از library خوانند از آن استفاده کنند

به خط این به اینترپت Interrupt های نرم افزاری System Call ها باید به جدولی compile می شود



Subject:

Year. Month. Date. ( )

معمولی منت درسته میشه چرا که به mode برای سیستم میرد

پسین رانجی

ماشین مجازی (Virtual Machine)

اول به سیستم ماشین رانجی و اتی جیت



از برکس جسی OS استفاده می کنند تا بتوانند  
از منابع استفاده کنند

(3)

Concurrent



چونیک application چند application به هم جیت  
همینطور حال اجراست یک در هر لحظه فقط یکی می تونه اجرا بشه

اگر دیدیم بهر حال در ماشین یه کاره کنیم تا خودمان رای سیستم و متوجه می بشوند که در منابع بهر میزان شریک نیست این  
به این کار می تونیم جسی است که OS اجرای هر یه این چند OS است

Application 2



این رانجی نیست بلکه رانجی (3) است

ماشین مجازی

به شریک منابع از دید بهر حال جسی است و چند OS  
و نتیجش به وقته ای در اجرای بهر ای می بشود بهر حال جسی علت شریک در منابع است  
متوجه این بهر میزان delay می بیند



## مدیریت فرآیند Process Management

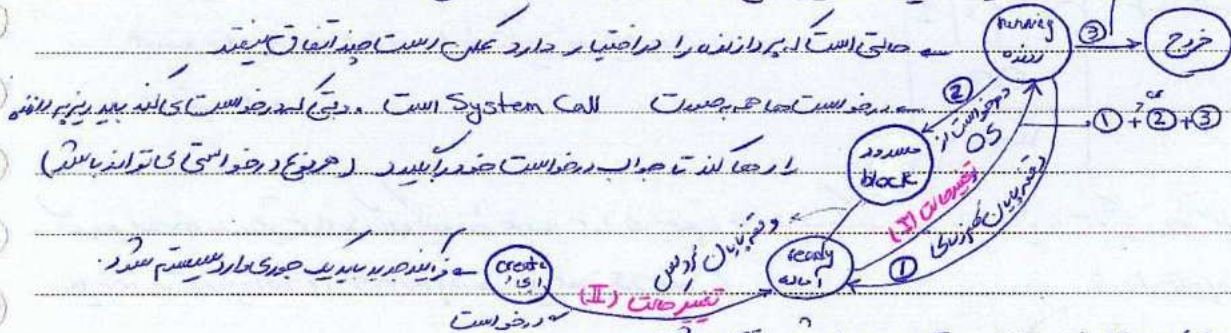
تجزیه فرآیند بر پایه درج اول اجرا  
فرآیندهای همزمان (همزمان)

زمان بندی (تسلیط بندی) فرآیند: فرآیندهای همزمان می توانند بر دانه را در اختیار بگیرند بلکه باید یکی یکی به دانه  
(روش جدایی از هم برای تسلیط بندی)  
را در اختیار بگیرند که در این جای الیوم تسلیط امکان پذیر نیست.  
از دیدگاه روش های زمان بندی، زمان این الیوم ها چگونه دست در خیر می باید به دست داده شود.

## حالت فرآیند Process State

فرآیند در طول زندگی خود در حالت های مختلفی قرار دارد این حالت ها تغییر می کنند

درخواست یک سرور از سرور خروج (exit) (درخواست تسلیط است)



در حالت خروج تمام منابع را از سیستم می رها می کند

در طی در حالت مسدود است برای اتفاقی که بعد از جواب درخواست آمده در سرور (مسدود می ماند) می تواند بر دانه

بر آید این به هم چون تسلیط می است پس باید در صف برود تا هر وقت تسلیط شود بر دانه را در اختیار بگیرد پس به

حالت آماده می رود تا هر وقت تسلیط شود بر دانه را در اختیار بگیرد و به running می رود پس به هم چون تسلیط می شود بر دانه را در اختیار بگیرد

بر دانه را در اختیار بگیرد تا زمانی که تسلیط شود بر دانه را در اختیار بگیرد

در تمام این حالت ها تسلیط می شود بر دانه را در اختیار بگیرد تا زمانی که تسلیط شود بر دانه را در اختیار بگیرد

را تسلیط می شود بر دانه را در اختیار بگیرد

چنانچه سرور در حالت تسلیط قرار دارد

تجزیه حالت ها به چهار دسته تقسیم می شود: ready, running, blocked, exit



دسته تغییر حالت ترمیم داده‌ها (تغییر حالت (I)

(II) ✓

تغییر حالت (I)

نمای هر حرکت از تغییر حالت جدید به حالت قبلی می‌باشد. در running این حالت را ترک کند و به حالت قبلی برگردد.  
این تغییر حالت اتفاق می‌افتد یعنی حرکتی از تغییر حالت‌های 1 یا 2 یا 3 اتفاق بیفتد.

تغییر حالت (II)

ایجاد یک فرآیند در خود است و فرآیند جدیدی را می‌سازد که فرآیند موجود در خود است. ایجاد آن به بعد. جمله ای در این  
یک System Cell است.

(point) ایجاد فرآیند توسط سیستم در خود است. فرآیند جدیدی است (توسعه فرآیند).

وقتی به یک فرآیند یک Icon، یک خطی که فرآیند را می‌نماید و یک فرآیند جدیدی که در آن قرار می‌گیرد می‌بینیم.

(point) ایجاد فرآیند توسط توسعه فرآیند جدیدی می‌شود و راه دیگری نیست.

این create نقطه توسعه فرآیند می‌باشد که فرآیند جدیدی را می‌سازد و می‌تواند به صورتی دیگر باشد.  
کاربری کاره است.

هنگامی که فرآیندی می‌سازد به نام فرآیند جدید را از دست به خود می‌دهد و در خود است create و آن این تغییر حالت است.  
حالت از دست دادن فرآیند می‌شود.

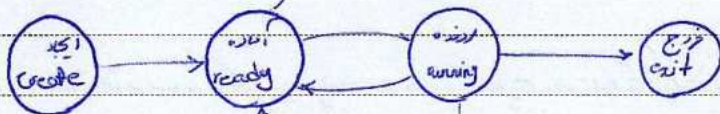
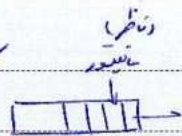
دانشگاه مهندسی کامپیوتر و فناوری  
اطلاعات دانشگاه صنعتی امیرکبیر



7, 19

جلسه هفتم

مکان است جدیداً فرایند در این جا باشد پس صبی این جا هست  
که نسبت در اختیار گرفتن برادرزاده را تعیین می کنند



وقته پایان درخواست

اینجا هم صدهای منتظرند تا جواب درخواست خود را بگیرند پس این هم  
صف داریم

بعضی از تغییر حالت ها ممکن است بر اساس حجم و وقته حجم درخواست باشد

تغییر حالت  
درخواست

سعی کنیم عامل توسط وقته یا درخواست به کاری افتد

ایجاد فرایند توسط واسطه فرایند دیگر، تماماً راه Active کردن یک برنامه منتظر نیست که یک برنامه که در حالت running است درخواست را بپذیرد

به عنوان نمونه برای برنامه های مختلف فرایند یا حالت اجرای در داریم، چه چیزی؟

یک فرایند داریم که دائماً در حال اجرا است فرایند ناظر یا monitor

فرایند مانیتور در دردی را چک می کند مثلاً وقتی یک خطی می بیند فرایند جدیدی مربوط به آنرا ای دی می کند  
خودش همین است و کار دیگر ندارد فقط فرایند ای دی می کند ای دی را بپذیرد و دردی چک می کند که برای هر چه  
چون فقط خطی ایجاد فرایند است پس اگر در حالت blocked است و خطی ای دی دهد  
بنا به monitor وقتی که در دردی آمد توسط OS بیدار کشیده می شود و بر بازنده مراجعتی قرار داده می شود

فرایند به صورت فیزیکی می چسبند به برنامه در حال اجرای می OS چگونه آنرا تشخیص می دهد؟

وقتی برنامه به فرایند تبدیل می شود OS باید متوجهی باشد که آن به چه ساختارهای داده ای برای آن نیاز دارد OS برای

هر یک از ساختار داده ای نیاز دارد و همین است که باعث می شود OS آنرا بشناسد. این ساختار را می نامند که به آن



process control block (PCB) (تجزیه‌ای است از داده‌ها). تازگی نه ساختار داده‌ای وجود دارد فرایند هم است از حد OS. PCB نماینده فرایند است. خروجی این PCB را ای‌دی‌لند وقتی حذف شد فرایند هم حذف می‌شود و منابع آزاد می‌شوند.

### داخل PCB چیست؟

حالت فرایند Process state

برنامه شماره program counter

رجیسترها CPU registers

نشانگر پشته stack pointer

اطلاعات مدیریت حافظه Memory management information

اطلاعات زمان بندی پردازنده CPU scheduling info

آیا اولویت دارد

2- درجه اولویت آن چیست

اطلاعات حسابداری Accounting info

تا به حال چند بار منابع استفاده کرده، چند زمان پردازنده را گرفته، چند بار خطا استفاده کرده

حوزه‌های که پردازنده را اجبار می‌کند و دلیل باید اطلاعات در پردازنده load شود تا برنامه پردازشی اجرا شود و وقتی هم

که می‌خواهد پردازنده را از برنامه بگیرد اول اطلاعات جاری را در PCB ذخیره می‌کند. پس PCB در هر زمان تغییر

می‌کند.

وقتی که PCB از پس‌ی‌ارود هر اطلاعات مربوط به برنامه از پس‌ی‌ارود پس به برنامه‌ای وجود ندارد پس برای پردازنده

که منابع را جدا کند از آن کنیم چون بهتری از آن منابع استفاده می‌کند پس از حذف PCB منابع آزاد می‌شود و حساب می‌شوند

point) اجرای فرایند = ساخت یک PCB

خروج از فرایند = حذف PCB



point) یک فرایند : که مرتب منابع از OS  
یک ساختار داده‌ای برای فرایند ایجاد می‌شود

فرایند اول چگونه‌ای می‌شود؟

چگونه‌ای از طرف "فرایند فقط توسط فرایندی می‌شود" نوشته می‌شود. فرایند اول دستی یعنی توسط کاربر ایجاد می‌شود  
(دستی ایجاد کردن فرایند - روشن کردن سیستم - reset کردن سیستم)

reset کردن (انجمنی): reset کردن یعنی به حالت عادی رجوع کردن (حالت عادی صفر است)  
که می‌شود صفر می‌شود. به صورت عادی به حالت عادی PC و SP و ... به حالت عادی می‌شود (حالت عادی می‌شود)

PC = ...

SP = ...

reg = ...

MMU = ... memory management Unit

ساخت یک PCB  
محصول دستی

دور از reset کردن این چه داریم یک PCB درستی می‌شود. همه اطلاعات را برای PCB قطع می‌کنند و  
اطلاعات لازم را تا به این می‌شود به حالت عادی می‌شود و در صورتی که reset کردن در صورت  
مستندات به درازن این می‌شود

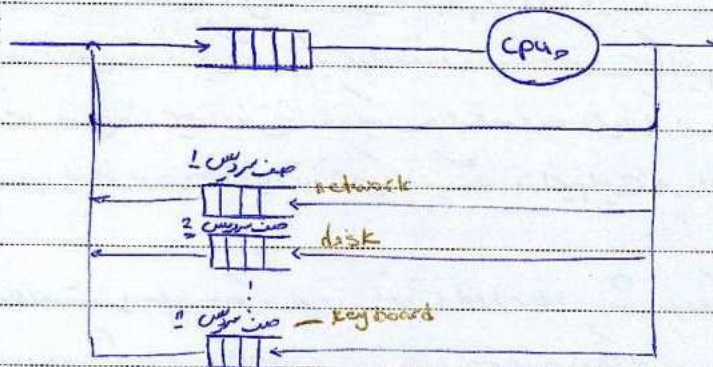
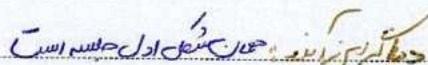
این فرایند اول اصلاً دستی از OS نیست چون این فرایند به درازن به درازن OS قرار می‌دهد و به این  
فرایند OS می‌شود که دور از این از BIOS است. یعنی دور از این OS را داخل حالت قرار می‌دهد و در آن  
کنترل به درازن به OS می‌دهد و در آن به حالت "فرایند توسط فرایندی می‌شود" و به درازن می‌شود  
OS همین فرایند است که توسط PCB که اول می‌شود و state می‌شود

PCB که اول به صورت دستی می‌شود و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود  
می‌دهد و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود

این PCB همین است که به درازن می‌شود و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود

OS اطلاعات دستی خود را در جدول می‌شود و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود  
PCB درازن OS وقتی در جدول می‌شود و به درازن می‌شود و به درازن می‌شود و به درازن می‌شود  
در این

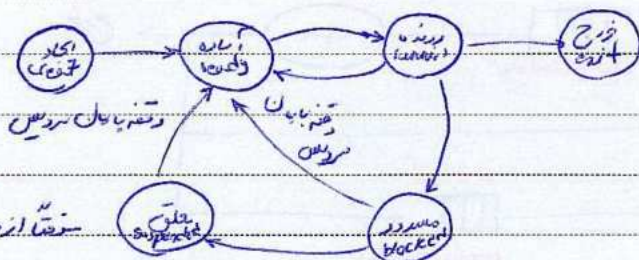


[illegible]

فردی و مستقل در داد و ستد، PCB

فقطی که نویسم فرایند در صف است پس باید همه اینها در این صف باشد و اگر اینطور نیست در واقع PCB را ننویسه  
فرایند است در صف است که تا زمانه این حجم که چون PCB بزرگ است تمام PCB در صف یعنی در یک صف فقط PCB  
در صف باشد کافی است





جلسه هفتم  
توضیح و رفع اشکال  
آمده به این جای میسر می شود  
۱۴۸۳۰۲۰۲

به ادماج برای برآیند خود در است <sup>محل حادثه</sup> <sup>بوی خود بر هم</sup> <sup>ی خاصه</sup> <sup>و نسبت</sup> <sup>کند</sup> <sup>کری</sup> <sup>خاصه</sup> <sup>اجماله</sup> <sup>جمله</sup> <sup>ساج</sup>  
 محدود است در حالت خلوت <sup>ی در در بی</sup> <sup>از</sup> <sup>نرایندی</sup> <sup>حسود</sup> <sup>در</sup> <sup>بود</sup> <sup>سفتا</sup> <sup>بیل</sup> <sup>ی</sup> <sup>الیم</sup> <sup>را</sup> <sup>صفت</sup> <sup>آنان</sup> <sup>بای</sup> <sup>و</sup> <sup>اعمال</sup>  
 که خواسته آن بر ساد <sup>خارج</sup> <sup>شکل</sup> <sup>خی</sup> <sup>دفع</sup> <sup>ری</sup> <sup>بیل</sup> <sup>چون</sup> <sup>روی</sup> <sup>اها</sup> <sup>ی</sup> <sup>که</sup> <sup>نانه</sup> <sup>ساده</sup> <sup>بلکه</sup> <sup>باید</sup> <sup>در</sup> <sup>بی</sup> <sup>نظیر</sup> <sup>شود</sup>  
 تمام تغییر حالت <sup>صدا</sup> <sup>باید</sup> <sup>بسیار</sup> <sup>در</sup> <sup>حالت</sup> <sup>یا</sup> <sup>دفعه</sup> <sup>توصیه</sup> <sup>لیم</sup> <sup>که</sup> <sup>ال</sup> <sup>تو</sup> <sup>لیم</sup> <sup>این</sup> <sup>ظرف</sup> <sup>بالا</sup> <sup>یا</sup> <sup>لیم</sup> <sup>حرف</sup> <sup>ی</sup> <sup>لیم</sup>  
 تقدیر است

این شکل تاریکی است و در اصل دهنده ای در قسمت blocked باعث می شود تا برای اینکه توانایی در حالت انتظار نداریم و هم در آینده در حالت آگاه ماندن و هم همین حالت ایستادیم پس وقتی ای در حالت blocked نیست ای توانیم حالت آن را تغییر دهیم و نتیجه حالت ای نداریم نیز می شود و هم

این راه حل برای سواری است هر چه فراموشی های در حالت مستعد و می شود

خروج از حالت مستودع حالت انقوع کی ایستد؟

درخواست جدید

۱۔ وقت پر پانچ سرورس

این بخش و بالا از ایندهای بسیاری را در آن راضی و آری مانند می همیشه جواب می دهد.







اگر زن آئیندار در حالت مستور و خلوت بود و در وقت پانچ سیزدهمین روز ماه در حالت آلوده بود و در حالت آلوده حتم  
کسی از مشرق حقیقت در حکم کسی از در حالت آلوده و خلوت چنانچه این زن آئیندار را باقیه ندارد باید مشرق چنانچه مشرق  
در حال آلوده و خلوت می رود اگر در این حالتی شود در حالت آلوده می رود

۱/ ترجمہ اس کی شکل  
۲/ فتنہ از جانب یحییٰ

صفت های ابتدایی که بتوانند روشی موجود باشند و در فرایند جدید عمل کنند (است چند فرایند)  
حجم state می شوند پس در خود است و صفت تدریجی می شوند به آنجا رسیدگی می شوند چه بسیاری از برای این  
صفت بندی پیش می گیریم (از اندیشم صفت بندی - زمان بندی - ترتیب بندی - scheduling)

point برای اینکه از حالت فاصله در به فاصله در بماند هم گفته است. پس چنگ الزامی نیست  
random اینجا که نام می برد پس به معنی نامی

صوبہ در حالات: بارہ است۔ ضلعی حکم است۔ رتہ بندی از برای دسی نظام سستیم و 05 کارکنہ ہا ہا صحت  
کوتاہ مدت کی کوئیز (short term queue)

mid term quize

long term queue,  $\tau_{\text{wait}} = \frac{1}{\mu - \lambda}$

الحمد لله الذي جعل هذا الكتاب من كتب الهدى والرشاد

Short term queue ← در زمان کوتاهی انتظار می کشد  
بسیار کمتری از سیستم به خواص اصلی می آید و مدت کوتاهی در آن می گذرد

nid term queue ← در وید های منتهی به یک جا که نشانی ی الیم ، در وید های منتهی به یک جا که نشانی ی الیم

long term queue ← دین صلہ زیادہ کا سرکشی کا افسیم

صفحه نوشته شده است: آنرا بنامه در حالت نوشته، همیشه از حکم زبانی ای را از نفس جوی حکم زبانی گرفته است. پس بنامه در آن  
مری زبانی که از حکم زبانی بنامه جدا کرده است حکم زبانی ضمایق است که حکم زبانی بنامه جدا کرده است و حکم زبانی  
است که در آن دو خواست مریدان است که حکم زبانی است که حکم زبانی بنامه جدا کرده است.



محرم الحرام ۱۲۸۵

سیستم های اول: <sup>۱</sup> مدل برای حفظ موجودات سیاسی - احای زمان بندی به کار برده و برای صف های بلند مدت و میان مدت از این سیستم ها استفاده می کنند چون نیاز روی سیستم زمان بندی ندارد و فقط برای صف های بلند مدت و میان مدت از این سیستم FIFO استفاده می کنند (محقق صف) چون عده کم است پس نیازی به کار روی سیستم ندارد.

First Come First Served

[illegible][illegible]

الموتى هم اهل بيته

(بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ) non presumptive

(new) preemitive

پردازنده را می‌توان گرفت. باید وضوح پردازنده را پس از یک لحظه در این آزمون‌ها می‌تواند. *non pre-emptive*

پردازنده را می‌تواند یکی از قضایا باشد. *pre-emptive*









نذر دہی -

نقشه

نقشه‌های استودیویی (ای) از نظر این جنبه برای سیستم‌های interactive کم نیست

ی سوالجہاں لکھی راہم اللہ صلوات : D:

7, 21 : 5 rules

الاولی تم های زمان بندی استراتژیها : (Process Scheduling)

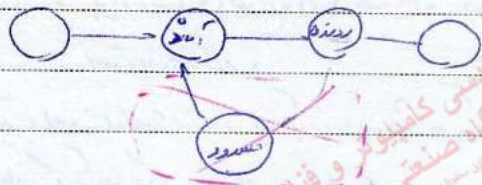
non-preceptive ← اینتدیری وقت مخصوص پر دانیہ و از نیست بدید

پیشہ ← preceptive در بیان امر و نهی

FCFS

E	D	C	B	A	فرایند های
8	6	4	2	0	تسلیع و حمل و نقل
2	5	4	6	3	تولید و توزیع (T <sub>1</sub> )
12	12	9	7	3	تولید و توزیع (T <sub>2</sub> )

روش اولی به روش فایو ایس FCFS است یعنی هرگاه درخواستی وارد شد در همان لحظه که می آید به آن عملیات داده می شود. این روش ساده است و هیچ اولی و برتری خاصی ندارد. این روش به سبب آنکه هر درخواستی که می آید به آن عملیات داده می شود و هیچ اولی و برتری خاصی ندارد.



20 18 13 9 3 0

[illegible]



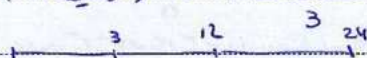
Substance	E	D	C	B	A	
2.56	6.0	2.40	2.25	1.17	1	( $T_T/T_S$ )

است

$\overline{T_S}$  :  
 زبان :  
 سردیس :  $\overline{T_T}$

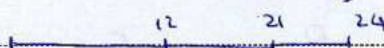
کرتب	A	3	3
ورد	B	9	12
✓	C	12	24

$(T + C_{\text{مطلوب}}) \rightarrow \underline{3 + 12 + 24} = 13$



ترتیباً ان (اعضای) داریم  $C \rightarrow B \rightarrow A$

$$\frac{24 + 21 + 12}{3} = 19$$



ایک حکیم نے (ترتیب) چوں کہ دستِ مرئوس کا حصہ و زوہد تر فرمائید گا اس پر دل کی فرستد  
دستِ مرئوس چوں کہ اپنے مرتبہ کا حصہ پس کیلیم جواب میں روشن بہ تقدیرِ مستحق دارد و تقصیری نہ کی  
خوب یادید دلِ مرئوس بہ دلِ مرئوس

اشکات - تفصیل و جوہر بنامہ

پیارے بھائی! سدا رہو کہ یہ صرف سدا رہے۔ یہ جس کی طرف اشارہ کرتا ہے

استفاده از سبزی

[illegible]

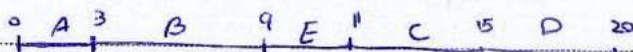


(Shortest Process Next) SPN 2

این روش گفته است  
در حال تپش این روش را استفاده می کنیم

	E	D	C	B	A	
8	6	4	2	0	نسبة	
2	5	4	6	3	نسبة	
11	20	15	9	3	نسبة	
7.60	3	14	11	7	3	$T_f$
1.74	1.5	2.8	2.75	1.17	1	$T_f/T_s$

در نقطه ه نقطه A است و دایره‌ای وجود ندارد در نقطه 3 نقطه B است پس باز هم دایره‌ای وجود ندارد در نقطه 4 در سه  
نقطه دیگری داریم که به این نام می‌گویند و اینها را انتخاب کنیم و به همین ترتیب  
برای انتخاب آنها را داریم و به این ترتیب می‌توانیم به این ترتیب انتخاب کنیم که به این ترتیب می‌توانیم به این ترتیب  
به این ترتیب انتخاب کنیم.



زمان پردازش از یک معلم است. معلم نسبت (ی) سیستم CPU از جای دانش آموزان به سیستم 105 از جای دانش

۱- اگرچه زمان مرگش را قبل از اجازتش می دانیم پس زمان کینه را اقلی است می آید و بسیاری را بدوی  
روشن های برای تخمین وجود دارد که محلی است از رانتهای خود باشد زمان مرگش را می دانیم خوش  
حیف نسیم می توانیم حدس بزنیم رانتهای ۱

[illegible]



Subject:

Year. Month. Date. ( )

1. point اصولاً الگوریتم‌های ادواتی دارای توانایی بهتری هستند

نوی‌ی‌الگوریتم محاسبات را حل کنیم. راه حل دومی مسئله است و اینست که الگوریتم را محض کنیم

حل مسئله ادواتی:

روش‌های برای تخمین وجود دارد که ما این حالت را برابر سیستم ریاضی می‌نویسیم

برای تخمین زمان بررسی

$$S_{n+1} = \alpha T_n + (1-\alpha) S_n$$

واقعیت اینست که یک فرایند بارها بررسی می‌شود و باید بررسی کنیم که این فرایند چقدر بار است و بررسی می‌شود. این تخمین را با استفاده از رابطه ریاضی می‌نویسیم

$$0 < \alpha < 1$$

S به تخمین

T به زمان واقعی

به این علت که الگوریتم زمانی که می‌نویسیم را ساده دارد.

$$S_n = \alpha T_{n-1} + (1-\alpha) S_{n-1}$$

برای سبب از فرمول اصلی الگوریتم برای اینکه بخواهیم تخمین بزنیم که چقدر بارها بررسی می‌شود

$$S_{n+1} = \alpha T_n + (1-\alpha) \alpha T_{n-1} + \dots + (1-\alpha)^n S_1$$

توان به باربرد اجماعی. اما اگر بخواهیم الگوریتم را بررسی کنیم تا ببینیم که آیا باربرد اجماعی است یا نه. اگر باربرد اجماعی باشد، الگوریتم را می‌توانیم به صورت زیر بنویسیم. اگر باربرد اجماعی باشد، الگوریتم را می‌توانیم به صورت زیر بنویسیم.

هرچه  $\alpha$  را بزرگ‌تر کنیم، به مقدارهای اخیر توجه بیشتری می‌کنیم. اگر  $\alpha$  را کوچک‌تر کنیم، به مقدارهای قدیمی‌تر توجه بیشتری می‌کنیم.

هرچه  $\alpha$  را بزرگ‌تر کنیم، به مقدارهای اخیر توجه بیشتری می‌کنیم. اگر  $\alpha$  را کوچک‌تر کنیم، به مقدارهای قدیمی‌تر توجه بیشتری می‌کنیم.



1/

Highest Response Ratio Next HRRN 3

استدلال برای انتضار  $T_w / T_s$

برای مثال

**P4PCO**



Subject:

Year. Month. Date. ( )

نمی‌توانیم به این ترتیب به جری قبلی بازگردیم چون این روش را از این به بعد نمی‌توانیم به جرم  
استفاده کنیم که از FCFS بگردد چون FCFS به معنای اولویت است و می‌تواند از این بگردد.

اگر می‌خواهیم از این روش‌ها non-preemptive استفاده کنیم انتخاب بهترین روش و به درج اولویت FCFS است  
چون SPN که می‌تواند به روشی باشد و HRRN هم دارای این ویژگی است و به درج اولویت باشد.

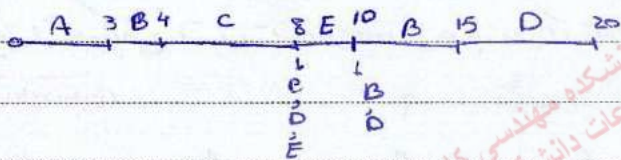
روش‌های preemptive :

4- SRT : Shortest Remaining Time (باقی‌مانده)

در هر لحظه می‌توانیم از این روش استفاده کنیم و SPN را اجرا می‌کنیم اما در هر لحظه اجرای آن

E	D	C	B	A
8	6	4	2	0
2	5	4	6	3

برای این روش 2، 3 تا انتخاب خواهیم داشت چون می‌توانیم به درج اولویت از این به بعد  
از این انتخاب کنیم در لحظه 3، B را داریم و در لحظه چهار هم B و C که B چون یک واحد از این به بعد 5 تا مانده می  
چون 5 تا برای C بماند و 5 است پس  
B تقاضای خود را C اجرا می‌کند.



انتظار داریم از SPN هم بگردد

در هر لحظه SPN را اجرا می‌کنیم و آن را به درج اولویت می‌کنیم و به اجرا می‌گذاریم.



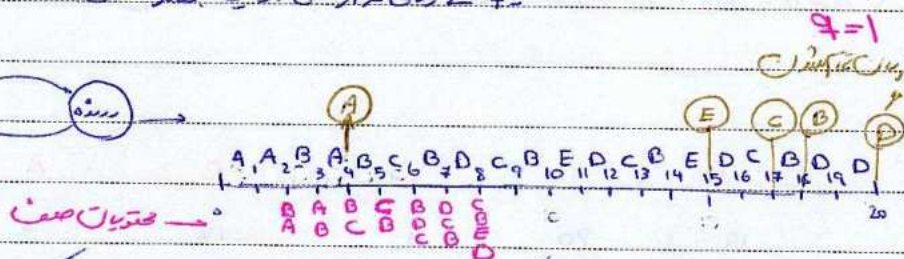
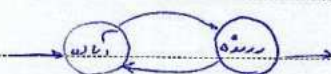
۵. موتی کھمڑی (موتی)

E	D	C	B	A
8	6	4	2	0
2	5	4	6	3

در دو حالت بررسی می کنیم ✓

1-  $q$  به عنوان نسبت کوچکی است ✓

2-  $q$  به نژاد از بعضی از اجناس است ✓



روزمان ۵. نقطه A است و کم خود برای تولید و به حالت آکنده می رود و می بیند و صفت کبی نیست پس در بار به حالت  
رونده می رود و یک اصل دیگر حالت را می آید که در نقطه 2. B هم آمده و این قرار را می آید و کم که در این حالت B بر  
صفت باشد و چون A همین الان سر صفت گرفته پس B سر صفت می آید و خاصیت کم را می بخشد و این کار را به نوبت آن  
است 1. اگر فرایند دارد شود و در همان لحظه فرایندی از حالت رونده به حالت آکنده می آید که بعد سر اول فرایند  
صفت و صفت تولیدی می آید.

سوالین	E	D	C	B	A	
	15	20	17	18	4	$C_{p,C_{v}}$
10.80	7	14	13	16	4	$T_D \text{ و } C_{v,i}$
2.71	3.80	2.8	3.25	2.67	1.33	$T_D/T_S$

درست نشود با اعداد بیشتر. صلی بهتر است و طبیعی است این اصل زبان بسیاری دارد و نسبت آن صد کرده و همین که روی  
 آن را در این آیه و در هر کجای زبان بسیاری می خواند.

[illegible]

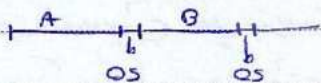


	E	D	C	B	A	
	19	20	11	17	3	$U_{24}, U_{25}$
10	11	14	7	15	3	$T_T$ $\phi$ $U_{25}$
2.71	550	280	1.75	2.5	1	$T_T/T_S$

اگر  $\rho$  از هم بزرگتر باشد، یعنی  $\rho > 1$  است، چنانچه  $\rho$  را اگر  $q$  را بزرگتر کنیم  
همان FCFS را میسریم حتی به کمترین اصله آن هم نزدیکیم

این خاصیت خود را است. برای اینکه بتواند همبسته با سایر SPN شود چون از یک الگوریتم استفاده نمی کنند

ماہی توالیم ۹ راضیہ کوچہ لیم چون سطح اس طوری نیست بلکہ دقیقاً سطح زیر است



در این مثال از زمان حساب می‌کنیم OS کارهای بلند مدت نظریه‌ی اول  
OS زمان بندی را ای‌ای می‌دهد نه CPU در نزد ترانه این کارهای بلند مدت کوچک است ولی همیشه می‌تواند در  
صورت تفرقه اگر 9 ضلع کوچک باشد مجبور هستیم زمان OS را هم محاسبه می‌کنیم