

Subject:

Year. Month. Date. ()

برای اجرای Thread ← نیاز به صفی می باشد و در حالت مشترک می باشد. نیاز به صفی می باشد تا بتواند کارهای خود را انجام دهد.

این کارهای خود چون در process هم هست
TLB برای به جای شدن ندارد چون فضای حافظه کش شده

افزایش دسترسی به Thread می چسبند و صفی random نیست
برای دسترسی می تواند Thread را شروع کند که هم compiler و OS این کار را می کند
آه به نظر می رسد که در این کار می کند

OS یک دست می رسد است و در داخل فرآیند کار می کند و در این می تواند بدون آن زندگی کند
که از طریق System call می توان این کارها را انجام داد

حالتی که Thread ها مشترک است و در یک Thread قفسی از حافظه را اختیار می کند و می تواند این
Thread برای ارسال محتوی این است که می تواند اطلاعات یک به یک را ارسال کند. نحوه ی ارتباط Thread ها از طریق
طریق فضای مشترک است. (که ارتباط process ها از طریق message است)

مزایای Thread

کمتر می تواند Thread برای اجرای کارها

امتیازاتی که در صفی می رسد و کارهای اصلی
حالتی که در صفی می رسد و کارهای اصلی
آه هر چه می تواند در صفی می رسد

مثلاً وجود caching و اطلاعاتی را از disk می آوریم در حافظه می توانیم در وقتاً چون می توانیم
در زمان اجرای برنامه و در هر یک از اطلاعات را Cache آوریم و می توانیم در وقتاً می توانیم از آن
استفاده کنیم پس به در وقتاً می خورد پس می توانیم صفی می رسد و کارهای اصلی

PC : Thread خصوصی

Stack ✓

register ✓
رجسٹری

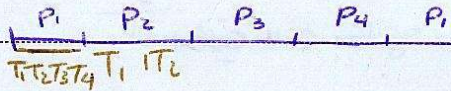
spread sheet ✓

دیگر خصوصیات

(Silberschatz , Stallings)

زمان بندی:

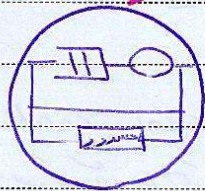
P₁ ... P_n



Thread خصوصی P₁ ... P_n ... Thread خصوصی ...
تقسیم ...
...

OS در وجه زمان بندی ایسی ...
process ...
Thread ...

process



OS این ...
...
...
OS ...

Subject:

Year: Month: Date: ()

point
برای برنامه‌نویسی جدید state machine. یک کارخانه ای که در اجرای یک برنامه دستور العمل تعیین شده است
در یک لحظه چقدر برای این دستور العمل را اجرا کند.

غذا کارخانه ای که برقی CPU ای که در دستور العمل ای که می‌شود مثلا توقف برنامه

اداره * می‌کند که به برنامه‌نویس اجازه داده شود که در وقت است. می‌تواند دستور العمل را برای حساب اصولی این

زمان اصلی طولی که می‌شود. عملیات سی‌پی‌یو floating point. عملیات طولی
جمله کوتاه

برای نوشتن این عملیات برنامه‌نویس می‌تواند CPU را در اختیار دارد چه دست‌خطی را دارد?

مثلا برای یک وب‌سایت. چه وقت در check کنیم آن‌ها می‌تواند به این حالت است که چک است
CPU (دانشگاه برنامه‌نویس) می‌تواند در این طولی است.

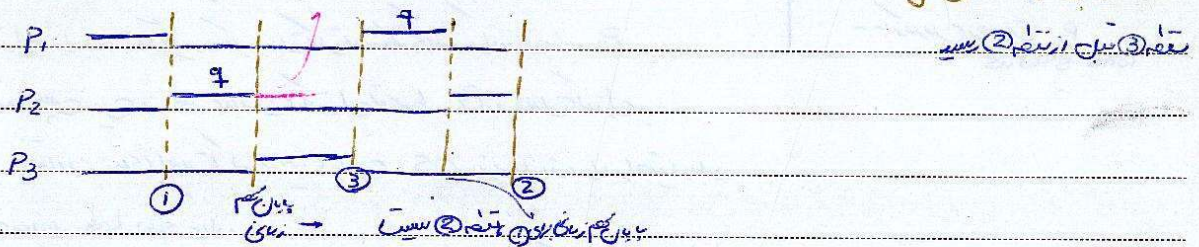
پس انتقال داده برنامه‌نویس از زمان طولی به کار می‌تواند به کاربردهای کاربردی (interactive) دستور العمل دارد

در کارهای غیر interactive دستور عمل دارد چون می‌تواند کار به برنامه‌نویس. می‌تواند با راضی کننده است.

تعداد اصل این دستور عمل Time sharing است؛ یعنی هر وقت خود است می‌تواند برای برنامه‌نویس را در اختیار می‌دهد در زمان

معمولی به تدریج می‌تواند تمام کند. بعد از دوره که برنامه‌نویس برای تصمیم (مستقران) سیستم عمل

Time sharing و دستور عمل. این دستور عمل برنامه‌نویس



همچون برای q (در این حالت می‌تواند کارها را) دست‌خطی

ای که می‌تواند

این دستور عمل را می‌تواند به دست‌خطی برنامه‌نویس به دست‌خطی برنامه‌نویس

در Time sharing می‌تواند به دست‌خطی برنامه‌نویس به دست‌خطی برنامه‌نویس

در این حالت می‌تواند به دست‌خطی برنامه‌نویس به دست‌خطی برنامه‌نویس

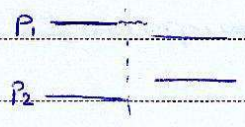
point در حالت Time sharing زمان می‌تواند به دست‌خطی برنامه‌نویس به دست‌خطی برنامه‌نویس

در n برنامه‌نویس

Subject:

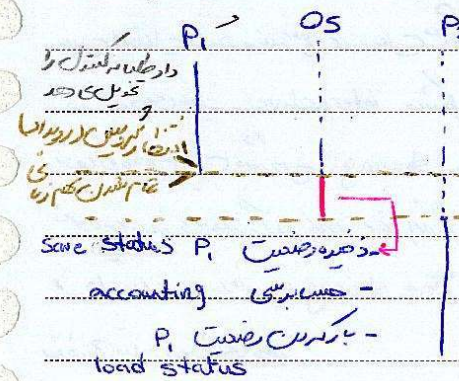
Year. Month. Date. ()

بین اجرای برنامه‌ها در زمان همزمان که در این سیستم عامل باید اجرا شود تا برنامه‌ها را از بی
 کلمه‌ها که در این سیستم عامل وجود دارد که چند رانندگی اجرا می‌کند
 بین زمان قطع برنامه و شروع برنامه‌ها که حاصل می‌شود



علاوه بر 6, 26

در سیستم عامل باید بتواند چند برنامه را به صورت همزمان اجرا کند و در این بین باید بتواند فقط یک عمل را در هر لحظه انجام دهد
 اجازت می‌دهد که هر یک از برنامه‌ها به نوبت اجرا شوند. چنانچه عملیات یک برنامه به نوبت اجرا شود در صورت طولانی در اختیار برنامه
 باشد و کار به نوبت برسد در این صورت برنامه‌ها می‌توانند به نوبت اجرا شوند



هرگاه از این اتفاق بیفتد
 P1 کارهای خود را تمام می‌کند و P2 شروع
 به کار می‌کند در این صورت برنامه‌ها می‌توانند
 به نوبت اجرا شوند و از P1 به نوبت به P2 کنترل خود را می‌بخشند
 این کار را می‌تواند OS این کار را می‌کند و OS در تمام این
 P1 کنترل را به دست می‌آورد و به نوبت کار می‌کند و کنترل را به P2
 می‌بدهد. پس بین این دو رانندگی کنترل خود را می‌بخشند
 P2 یک انتقال برای این است که در این زمان OS نیز برنامه‌ها را در اختیار دارد
 و کنترل کار switch می‌کند

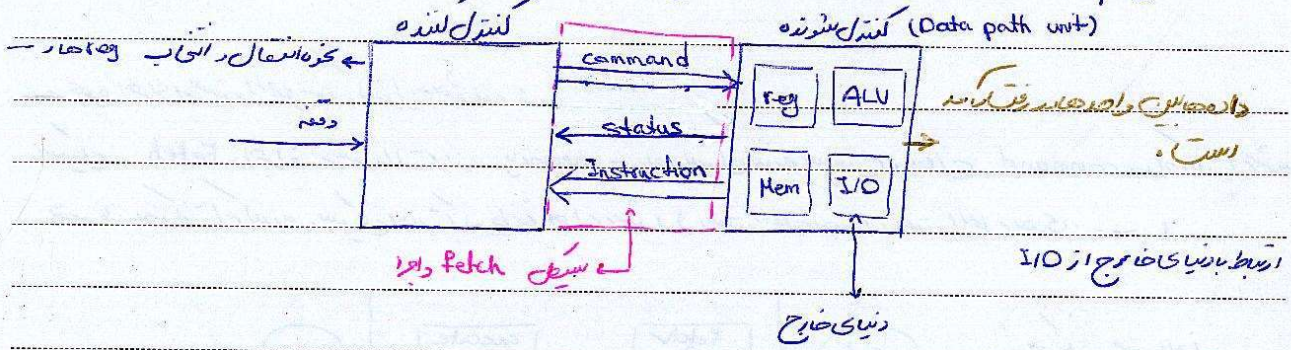
OS به نوبت به دست می‌آورد و P2 کنترل را به P1 می‌دهد و این کار را می‌کند
 می‌تواند که این کار را می‌کند
 OS تا برنامه‌ها را در اختیار بگیرد هیچ کاری از آن مسافه نیست پس OS همیشه تا زمانه اجرا می‌کند
 اجرا می‌کند

Subject:

Year: Month: Date: ()

معماری سیستم های دیجیتال

معماری سیستم (مختار) کامپیوتر از دو بخش تشکیل شده:



کنترل کننده و کنترل شونده: در دستور العمل ها، دستور العمل ها را می خوانند و command های فرستاده، دستور العمل ها را memory هستند. کارهای جدیدی خوانند و خود را در command های جدید می خوانند. برای فرستاده علاوه بر دستور العمل ها باید از وضعیت (status) مطلع بشود (Zero, overflow, carry).

دفعه: سیگنال های اندک و دفعه (تایمینگ) بر زمان و انتقالاتی که در مدار و دوباره برنامه منطقی را اجرا کنند.

- واحد کنترل کننده ← hard wire ← کمتر است.
- microprogram (از بین رفته)
- به خاطر سیم کارت یا سیم از بین رفته

reg ها: → رویت شونده (observable) → هنگام اجرا، این سیگنال قابل استفاده بود.

Stack pointer, address reg, Data reg

رویت نشونده (nonobservable) → برنامه نویسی از این ها می توان استفاده کند. برنامه نویسی که سیستم را کار کرده می تواند از این ها استفاده کند. کنترل های خاص و وضعیت.

دیده شدن و شنیدن از دید برنامه نویسی نیست

التر سیستمی، بشود بدون OS. ... حالتی قابل از reg های رویت نشونده کار کرد. reg های رویت نشونده و سیستم عامل این ها می خوانند (وضعیت از فرستاده می خوانند تا بعد از آن کار کنند).

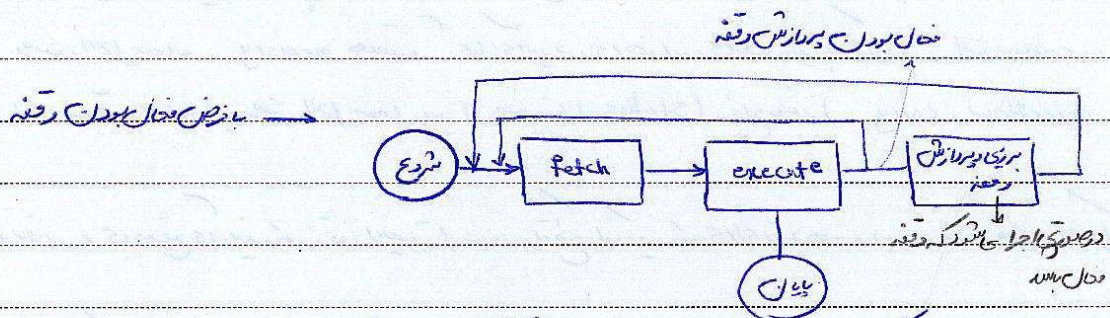
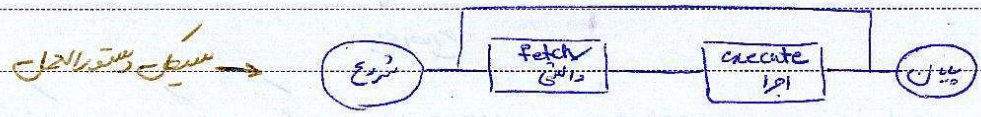
Subject:

Year. Month. Date. ()

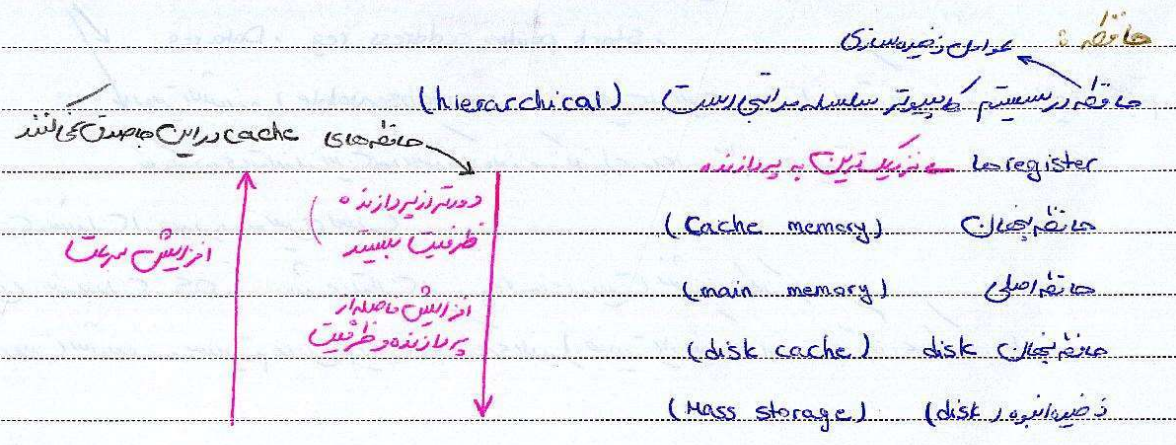
چگونه ممکن است این reg ها برای OS دیده بشوند و برای P, P2, P3 ... منتهی چه نسبتی از این می باشد
که ام برنام است که اجازه دهد چه کارایی پیدا کند؟

نوعه اجرای دستورالعمل در کنترل کننده؟

سیکل Fetch و اجراء دستورات از memory به حافظه کنترل کننده سپس ارسال command به واحد کنترل کننده
توسط واحد کنترل کننده. سیکل است که طی آن اجرای دستور (دستورالعمل بجای، دستورالعمل بجای) ...



در پایان اجرای دستورالعمل یک سرخطی به دقیقه داریم که به اینم چیزی اضافه یا نیامده که ایندوره بعد دستور بجای می آید
بردارش دقیقه و بعد از اجرای دستورالعمل در حافظه سیکل Fetch اجرا می شود
بردارش دقیقه های خارجی در پایان اجرای دستورالعمل (دقیقه داخلی هم داریم)
که دقیقه ها نباید امکان پیاده سازی هم برای وجود ندارد



Subject:

Year. Month. Date. ()

حافظه‌های سلسله مراتبی ← علاوه بر بلاک‌های فرمت‌یافته هم بلاک‌های بدون فرمت‌یافته (raw) داریم
در register (حافظه) ← حافظه‌های کوچک‌تری داریم که در register قرار می‌گیرند

← حافظه‌های کوچک‌تر را چه می‌نامند؟

register و cache (حافظه‌های کوچک‌تر)، خود برنامه‌نویس (در زمان اجرای برنامه)

حافظه اصلی: سیستم عامل (در زمان اجرای سیستم عامل)

disk: سیستم عامل ← در زمان اجرای سیستم عامل، block‌ها تقسیم‌بندی می‌شوند برای file system (مانند FAT)

حافظه‌های cache: خود برنامه‌نویس، حافظه‌های کوچک‌تری که در OS قرار می‌گیرند

اصلی از خود برنامه‌نویس و چیزی که در cache قرار می‌گیرد، حافظه‌های کوچک‌تری که در زمان اجرای برنامه

برای برنامه‌نویس، حافظه‌های کوچک‌تری است

حافظه‌های disk: حافظه‌های کوچک‌تری که در OS قرار می‌گیرند، حافظه‌های کوچک‌تری که در زمان اجرای برنامه

قرار می‌گیرد، حافظه‌های کوچک‌تری است

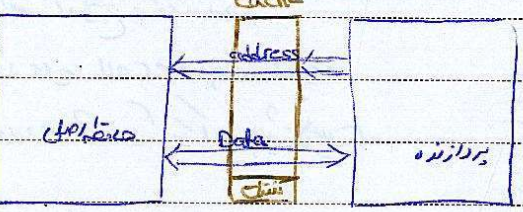
← OS در حافظه اصلی با disk برای پیدا کردن داده‌ها در حافظه اصلی کار می‌کند

حافظه اصلی

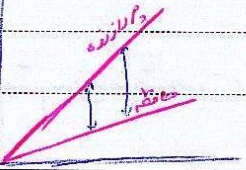
سیستم عامل برای کار خود نیاز به حافظه دارد و این حافظه را در حافظه اصلی قرار می‌دهد

حافظه‌های کوچک‌تری: cache

برای کارهای کوچک‌تر، حافظه‌های کوچک‌تری در حافظه اصلی قرار می‌گیرد



حافظه‌های کوچک‌تری در حافظه اصلی قرار می‌گیرد



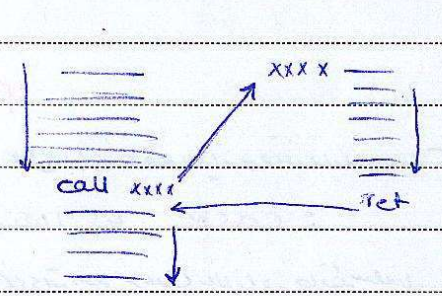
Subject:

Year. Month. Date. ()

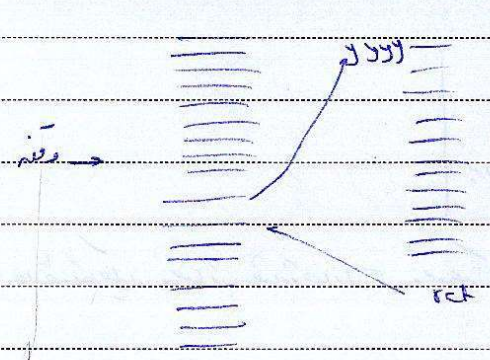
پیدا کننده در اصل آدرسها را برصافه اصلی می دهد و خط آدرسها برای آن فرستاده می شود. cache در این
 وسط قرار گرفته و آدرس را نگاه می کند تا این آدرس را در این آدرسها در این آدرسها در این آدرسها در این
 می فرستد. همچنین می تواند در cache می شود. در این آدرسها در این آدرسها در این آدرسها در این
 در سرعت بالا و حافظه کم (locality) هم از نظر زمان هم از نظر مکانی می دهد.

Temporal → آدرسهای که اخیراً در حافظه دسترسی پیدا کرده احتمالاً باز هم دسترسی داشته باشند (مثلاً در برنامه ها)
Spacial → آدرسهای که در کنار هم قرار می گیرند احتمالاً به هم دسترسی پیدا می کنند
 همچنین این است که لازم نیست طریقی را در cache داشته باشیم بلکه به دلیل اصل کلیت نسبت به cache
 دسترسی داریم و حداقل سرعت 95٪ است.

cache را تحت اثری که برای این کاربرد است. همه آن دست انداز می کند. بخش اصلی کار این است که
 ای می دهد. آنکه آدرسها را در این Data را قطع می کند. به طوری که آن آدرس را بهیچ وجه در دسترس
 چیزی را از ظرفیت خود می برد و آن را با سایرین می کند. تمام این کارها را پنجره از OS و CPU ای می دهد.



مدت زمان در حافظه
 return و call



مدت زمان در حافظه
 return و call
 در این حالت هم مثل همین است و هیچ فرقی ندارد
 فرقی ندارد با وقت که می گذرد. call معلوم است و می
 در وقت هر وقت که می خواهد وارد می شود و به آدرسها که می فرستد
 می رسد. این است که می رسد.
 هر وقت وقت می رسد می تواند به آدرسها را قطع کند

این که می رسد وقت برای سیستم عامل می رسد است.

← خطا OS بر اساس وقوع کار می کند (به عنوان قوی است)

دسته بندی وقفه ها (interrupt & exception)

1- اجرای برنامه: مثلا برنامه در حال اجراست و عملی ای می باشد که باید انجام دهد (تقسیم حافظه، دستکاری از دستورات عملی)

تجرب شده: به عنوان overflow در خروجی از فضای تعیین شده (برنامه باید خطا کند برای اینجا باید چه کار کند؟) تا در واقع برنامه در این مواقع تولید وقفه است و تولید وقفه باعث قطع برنامه می شود و به جای اجرای مجدد (تولید وقفه) کار می کند و به این ترتیب وقفه است که کار را انجام می دهد. با توجه می کند به برنامه را قطع می کند (برنامه هم چنین می شود برای ای) این کار ندارد و باید برنامه تمام این کارها را انجام دهد و در صورت اجرای دستورالعمل قطع می شود

← عکس العمل برنامه در برابر خطا یا عملیات غیره از (تولید وقفه)

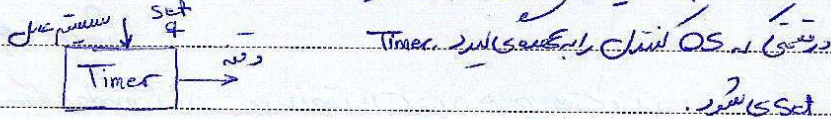
2- زمان اجرا (Timer): هر برنامه ای که می تواند زمان را در نظر بگیرد و وقتی زمان اجرای برنامه تمام شود تولید

وقفه می کند و به این ترتیب عمل تمام می شود

← کمزری برنامه ای که در این (نوع اجرا) عملی است که آن را در زمان مشخصی Set یا نیم برنامه

کمزری (19) در واقع کار می کند و هر وقت زمان 9 رسید تولید وقفه می کند OS این را

Set می کند و در این که کنترل را به برنامه می دهد و Set می کند



در وقتی که OS کنترل را به برنامه می دهد. Timer. Set می شود

3- ورودی خروجی IO: هر سیستمی که می تواند با IO کار کند. mouse



استطاعت با ضایعات خروجی از خروجی این port ها است که برای ورود وقفه دارند

برنامه ای که منتظر اطلاعات ورودی است از طریق این اطلاعات وارد شده و این کنترل را

برنامه OS است. ورود اطلاعات خروجی باید صورت گیرد و در این اطلاعات

وقفه ای صادر می شود و به این ترتیب برنامه در حال اجراست که این راه های از ضایعات و ضایعات ورودی را می برد و برای

از ضایعات و ضایعات برای برنامه منتظر است اما در هر دو وقفه های IO به این عملیات ورودی خروجی را اعلام

می کند. به این ترتیب اطلاعات کار می کند و در هر دو وقفه ای که out می شود

Subject:

Year. Month. Date. ()

آخرین دفعه سیستم عامل ریفرش کردیم و این سیستم ۵۰ مینوت طول کشید.

۴. توضیح مختصر از انواع (HW Fault) می توانند طوری ساخته بشوند که در صورت خراب شدن تولید دفعه کند
ساخته شدن راه برای صاف کردن و دوباره parity که در صورت مشابه تولید دفعه کند
تفصیلاتی می توانند طوری طراحی شود که تولید دفعه کند تا خطای مشخصی خرابی های تکراری دفعه ای تولید کند به اطلاعات کم را
فضیله دارد. هر مشکل کن از برای می توانند این دفعه را تولید کنند به طوری که این طوری ساخته شده باشند.

← دفعه نوع اول در صورت اجرای دستور العمل می تواند به نایب را رفع کند چونکه دستور العمل دیگری نیست و اصلاح می تواند
آن را انجام دهد و در سایر دفعه ها دستور العمل تکمیل اجرا می شود
در نوع اول دفعه نباید دوباره دستور العمل قبلی که در حال اجرا است در سایر دفعه ها چون دستور العمل کامل
اجرای دستور العمل می شود به دستور العملی که در حال اجراست و در صورتی که این دفعه را اجرا می کند.

← برنامه های کاربردی نباید بتوانند با port ها کار کنند و باید از سیستم عملی که صادر این کار را انجام دهد برای اینکه
حواسش که صادر این کار کند هر چه در دسترس است OS این را می تواند مدیریت کند. چندین بار port طوری که قبلاً از
رسیدن اطلاعات کم از برای تمامی دستور العملی که صادر شده است به این کار باید رسید OS باید چگونگی رسیدن
را مدیریت می کند هر اطلاعات دردی را که می خواهد به این که در صورتی که این می تواند جلوی کار را بگیرد پس باید طوری
کنیم که نفیض نباشد.

دو حالت (Dual mode)

پس مانده دو mode اجرایی دارد به اصطلاحات معروف به برنامه های می توانند طوری که کارهای دیگری را تولید دفعه
اصطلاحات زیاد به هم می رسند و در هر کاری که می توانند
System mode : اصطلاحات زیاد دارد (مستقیمی به Sct, port کردن Timer)
User mode : اصطلاحات کم دارد

Subject:

Year. Month. Date. ()

مطلبه چهارم:

با port های ورودی خروجی از طریق دستورات ورودی خروجی سیستم کار کنیم و اداره port های OS است
 در هر زمانه ای که می‌خواهیم port ها کار کنند باید از سیستم عامل بخواهیم.
 در هر سیستم می‌توانیم با port ها reg های که می‌خواهیم کار کنیم در همه طایفه‌ها به دست آوریم این کار را با سیستم
 در همه طایفه عملیات غیره که توسط وقفه قطع می‌شوند و بعضی کارها به خصوصه‌های در OS است

برای طایفه‌های جدید باید بیت اضافه به نام Mode bit به سیستم

کامپیوتر

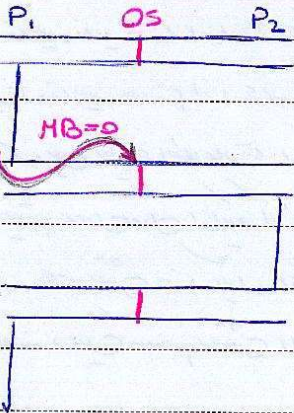
بیت به دستورات عملی در opcode اضافه می‌کنیم تا Mode bit را نشان دهد



در اجرای دستورات عملی نگاه می‌کنیم اگر بیت اضافه را در دستور داریم
 دستور را عملی می‌کنیم در غیر این صورت آن را نادیده می‌گیریم

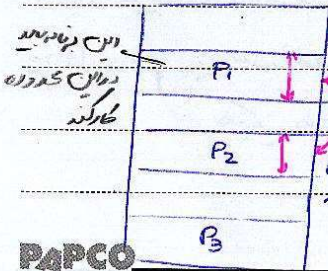
Mode bit چگونه تغییر می‌کند

در شروع کار MB باید صفر باشد (در reset و اختلالات سیستم و زیاد و 0 بکار می‌آید چون شروع کار OS است)
 حالای خاصه که می‌خواهیم را به برنامه‌های دیگر می‌دهیم قبل از اینکه برنامه



کمی در دستورات عملی که در بعضی کدها Mode bit از دستورات OS
 است و وقتی در دستورات عملی که در بعضی کدها Mode bit از دستورات OS
 برنامه‌ها را می‌خواهیم Mode bit را صفر کنیم تا قطع می‌شود

حفاظت حافظه (memory protection)



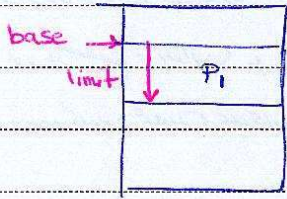
چندین برنامه داریم که می‌خواهیم اجرا می‌شوند و این برنامه‌ها باید برای اجرای حافظه اختصاصی باشند
 چون برای اجرا PC باید به کل برنامه‌ها دسترسی داشته باشد و این کار را می‌توانیم با حافظه اختصاصی
 می‌توانیم با حافظه اختصاصی که می‌خواهیم در کدها اختصاصی می‌دهیم و می‌توانیم این کار را با حافظه اختصاصی

P4PCO

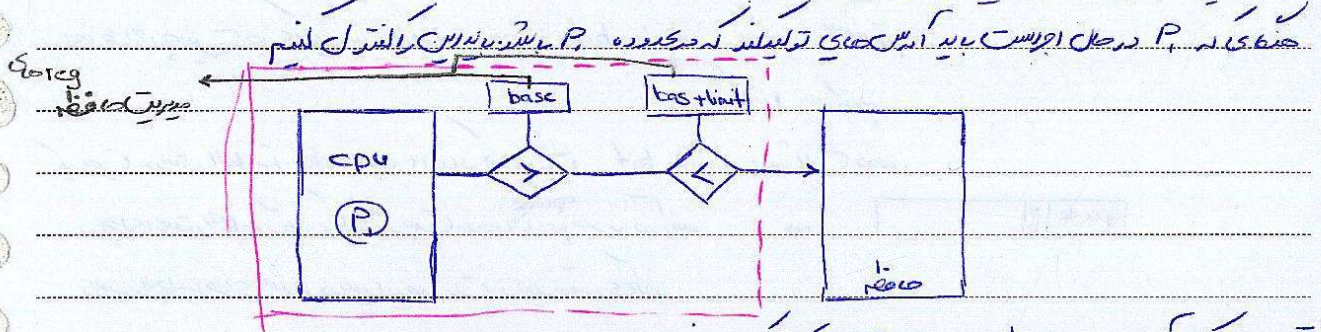
Subject :

Year . Month . Date . ()

بسیاری از پردازنده‌ها چون باعث تغییر data می‌شوند (بنا بر وضعیت) باید جلوی آن را بگیرند (ممنوع است)
ممنوع است. هر کس در کرده خود را بتواند بکاراند



هر پردازنده یک آدرس base دارد که محل شروع برنامه و یک limit دارد که طول برنامه است. هر پردازنده با این دو عدد در اختیار OS است (ممنوع است) یعنی برنامه OS نیست و می‌تواند رانند که در اختیارش و در این اصل است (ممنوع است) در این جا جلوی کارهای غیر مجاز این پردازنده را گرفته است



قبل از اینکه آدرس با به حافظه بچشم دست شرط با چک کنیم تا بچشم در کرده است یا نه ؟
1- باید حتماً پردازنده از base باشد
2- باید که کمتر از base + limit باشد تا در کرده P1 باشد

این مقایسه باید تحت نظارتی انجام شود و مقایسه کننده در واقع این مقایسه کننده داخل CPU است
آنگاه این مقایسه همانم نظارتی بر روی سیستم دیتا پردازی صورت می‌گیرد. کار OS در این مقایسه base و limit را
تایید کرده یعنی این اطلاعات را داخل reg (دیتا نوشته شده) گذاشت
سیستم عامل قبل از اینکه کنترل را به P1 بدهد base و limit مربوط به P1 با کاری کند و بعد کنترل را
بدهد که نقش ضد ایمنی دارد
ممنوع است بر این حجم نقش دارد که کاره این reg ها (چون دیده نشوند هستند)

دسترسی‌های مربوط به reg های خاص از نوع ممتاز (privilege) هستند
privilege ← دسترسی‌های خاص که در سیستم قابل اجرا هستند و در سایر اجزای سیستم و تغییر bit Mode
همگی دسترسی‌های privilege است

Subject:

Year: Month: Date: ()

به طور کلی برنامه‌ها فقط دستورالعمل‌های تعیین شده اجرا می‌کنند و می‌توان آن را طوری طراحی کرد که در زمان کار خودی که می‌باید انجام دهد تولید وقفه کند پس از آن به سر و پس از وقوع آن وقفه می‌رود و ... یعنی جوابگوی آن در شرایط اضطراری به صورت تولید وقفه است

← سیستم عامل در صورتی که توان خود را از دست بدهد که خطای سیستم قوی وقفه باشد.

← در چه حالتی P کنترل بر OS از دست داده و OS به دست افتاد؟

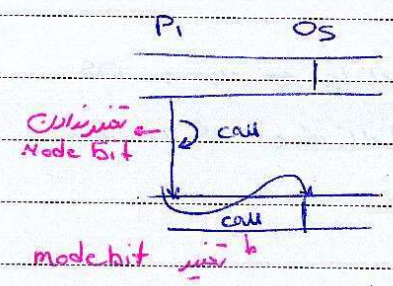
در حالتی که در طول اجرای برنامه خطای رخ دهد یا چیزی از OS بخورد و مثلاً دردی یا خطای چیزی از سیستم رخ دهد سیستم عامل را تا حدی که خطای آنست و در صورتی که سیستم عامل در این وقفه که خطای وقفه بجم زبانی است و وقفه

وقتی که در زمان اجرای برنامه از OS کرد و اجازه می‌دهد تا زمانی که برنامه به سر و پس از در صورتی که خطای رخ دهد تا برنامه اجرا شود و در صورتی که جواب در صورتی که کنترل بر آن برنامه خطای رخ دهد

point) برنامه‌های وقفه‌خیز OS هستند و در وقفه OS اجرا می‌شوند

← چگونه می‌توانیم در صورتی که وقفه رخ دهد؟ امکان دارد با call یا continue

این call به معنی call است و در حالتی که در چه جا به پیوسته call خوانده شود و در زمان هستند Mode bit را عوض می‌کنند و این call برای در صورتی که Mode bit را عوض کند.



System call ← می‌تواند از call خوانده برای OS هستند و در این حالت

شده اند (نمی‌تواند از فراخوانی‌های تعیین شده برای OS)

نوعی از call قوی و System call به طوری است که می‌تواند از آنرا استفاده در System call

Mode bit را تغییر دهد.

Subject:

Year. Month. Date. ()

نکته ۱. Intel برای این کاربرد دستورالعمل جدید کرده تا هم از Interrupt که دستور داده و از ما بعد صورتی کند

INT XX (58)

Intel یک صیقل INT را دارد به عنوان 58 (مدرسه صوری INT است) کاربرد آدرس را برای بار

دادن به برنامه را از این اجرای کند پس همان که هست در این حالت به در این غیر مستقیم است

(چونال فقط عملی کار کرد در اختیار سیستم عامل هستند)

مفید این اجرای شده طور اثر این mode bit صوری شود و می بیند کنترل را به برنامه بر می گرداند به OS

درخواست از OS ← System call ← تغییر mode bit صورت می گیرد (در برنامه سازی دستور این تغییر

شده و نیاز به دستور اجرای تغییر mode bit نیست)

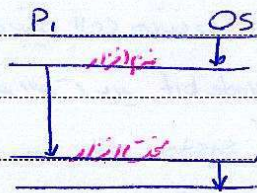
تغییر حالت کاربر سیستم مده از طریق Interrupt (مده ای می شود) (محل اجرای شد ID)

در واقع در انتقال این P کنترل را از دست داده می گویم (با مده)

تغییر mode bit با رفتن از مده کاربر سیستم از طریق کن ابزار است

Interrupt یعنی تغییر mode bit می شود

رکورداتیست و صورتی از ابزارهای این کار



OS ← برنامه ← نرم افزار

نرم افزار ← OS ← سخت افزار

Subject:

Year. Month. Date. ()

معماری سیستم:

- پیمان نامه طراحی اجرا و دستوراتی که در آن آمده
- در برنامه ای که کاربر می کند در مورد پیمان نامه را در اختیار بگیرد و در مورد اختیار و محدودیتها را بداند
- یعنی از امنیت سیستم از دید OS که می خوانند و در آنرا کنترل می کنند مثل حافظه مکان یا cache

اجزای سیستم عامل:

- process management مدیریت فرآیند
- memory management مدیریت حافظه
- Secondary memory management مدیریت حافظه ثانویه
- Input-Output management مدیریت درستی خارجی
- File management مدیریت فایل
- Networking که در آنجا بین سیستم های سیستم شبکه سازی
- command Interpreter System سیستم دستورالعمل ها
- جزایر OS هستند که در کنار OS است

مدیریت فرآیند:

- برای اطلاع و در دسترس بودن
- برنامه ای که در حال اجراست
- active
- برنامه ای که در حال اجراست و در دسترس است
- process یک چیز است که در حال اجراست و در دسترس است
- اجرا در هر یک process می شود
- program اینگونه نیست که در حال اجراست
- مدیریت فرآیند یک فرآیند است که OS ای که در دسترس است
- آن کار دارد

مدیریت فرآیند در حالت تعلیق و بیدار شدن:

- 1- ایجاد صفت فرآیند تعلیق → فرآیند که در دسترس نیست
- 2- بیدار کردن فرآیند در حالت بیدار → فرآیند که در دسترس است
- resumption suspension

Subject:

Year. Month. Date. ()

برای اینکه بتوانیم فرآیندها را به شیوه صحیحی مدیریت کنیم باید بدانیم که این فرآیندها چگونه می توانند به هم دسترسی داشته باشند و چگونه می توانند به هم دسترسی نداشته باشند. در اینجا به بررسی فرآیندهای متعلق به سیستم عامل می پردازیم.

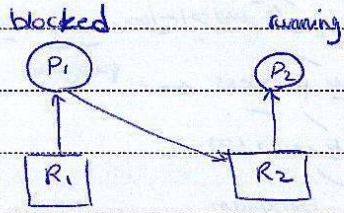
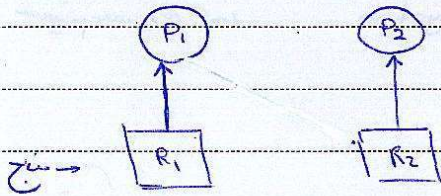
synchronization

3- وظیفه اصلی برای همه فرآیندها این است که در زمان اجرای خود به هم دسترسی نداشته باشند. این کار را می توانیم با استفاده از ابزارهایی مانند semaphore یا mutex انجام دهیم. این ابزارها به فرآیندها اجازه می دهند تا در زمان اجرای خود به هم دسترسی داشته باشند.

4- وظیفه اصلی برای فرآیندها این است که در زمان اجرای خود به هم دسترسی نداشته باشند. این کار را می توانیم با استفاده از ابزارهایی مانند semaphore یا mutex انجام دهیم. این ابزارها به فرآیندها اجازه می دهند تا در زمان اجرای خود به هم دسترسی داشته باشند.

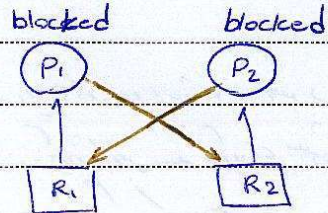
Dead lock

5- در این حالت برای حل مسأله بین سیستم و فرآیندها این روش را پیشنهاد می دهیم. در این روش هر فرآیند (P1 و P2) به یک ریسورس (R1 و R2) دسترسی دارد.



در این حالت P1 تا زمانی که R2 را نگه دارد نمی تواند به R1 دسترسی داشته باشد. در حالی که P2 در حال اجراست و به R2 دسترسی دارد. این حالت می تواند منجر به Dead lock شود.

این حالت از OS اجازه می دهد تا فرآیندها را مدیریت کند. در این حالت هر فرآیند می تواند به فرآیندهای دیگر دسترسی داشته باشد.



وقتی P1 و P2 به فرآیندهای دیگر دسترسی نداشته باشند، این حالت می تواند منجر به Dead lock شود. در این حالت هر فرآیند نمی تواند به فرآیندهای دیگر دسترسی داشته باشد.

Subject :

Year . Month . Date . ()

این که OS بجز این حالت اتصال استفاده کار راهی نیست یعنی اگر به هر نحوی ممکن است راهی در دسترس باشد
گویی که این راهی تا آنکه راهی ارائه دهد OS است چون چند P_1 و P_2 خواب هستند (غیر فعال) پس
می توانند برای ایام دهند

مدیریت حافظه linear array

حافظه به صورت فیزیکی یک آرایه خطی از قطعات است
که از آنجا می توانیم استفاده کنیم (استفاده کرد) بعد از هر استفاده ای توان استفاده کرد

نوع های مدیریت حافظه

1. **کی لبری** : قطعات استفاده نشده در حافظه آن یعنی با استفاده از تقسیم ها از حافظه استفاده نشده در حافظه آن می
است (ضد OS این کار را ای ای می کند)

2. **تخصیص حافظه** : پس کی لبری آن

3. **کی لبری** : فضای آزاد پیدا می کند که فضای آن است تا آنکه خطاها را بگیرد (یعنی حافظه به خود کار را به خود

مدیریت حافظه دیسک (Disk)

صحنی به شکل مدیریت حافظه است

1. **مدیریت فضای آزاد**

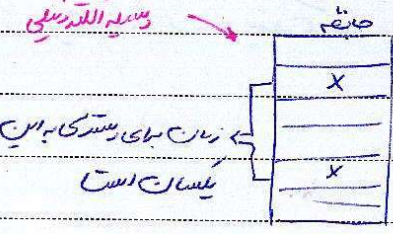
2. **تخصیص مکان ذخیره سازی**

3. **زمان بندی (مانند کی لبری) دیسک** : تقاضاهای برای دیسک باید به ترتیب انجام شوند (یعنی به ترتیب درخواست را اول ای)

دیسک این را در این جا مطرح می کنیم حافظه یا دیسک فرکانس دارد و تمام قطعات یکی آن یکسان است

scheduling

دیسک یک وسیله مکانیکی است جدول جدول تقویم است همین جدول می شود
مکان ها به هم فرکانس (مکان) نزدیک به هم در زمان است و زمان را با هم می توان
رشته ای (مکان) به هم می رسد (تبدیل) می شود



Subject:

Year. Month. Date. ()

هاتف ، زمان دسترسی مکان های مسافر است

دسترسی ، زمان دسترسی مکان های مسافران است .

برای اینکه در کمترین زمان دسترسی را ایجاد کنیم از قابلیت بندی استفاده می کنیم مثلا اول به بندتهای پهن می رویم که مکان آن به بندتهای مسافر است

میزبانی دوری خردی 1:

یکی از کارهای مهم 50 management میزبانی مسافر است

1- میزبانی مسافر ها ، برای اینکه بتواند بر روی کار را اجرا دهد

2- کنترل دوری کار خردی ها : مثلا می خواهیم با port دوری ارتباط برقرار کنیم. قبل از این لازم است port را چک کرد و اگر می خواهیم خود را به port وصل کنیم باید هر چیزی را در مورد port بدانیم و می توانیم کنترل ایضا را برای کار کنیم

3- سیستم بازی : این چیزی شبیه جنته cache است که می خواهیم با port کار کنیم با بازی کنیم چون port ها لازم است در وسط این port و پیرامون هستند و می توانند cache ها کنترل کنند از برای اندازه پس کار اصلی بازی بر روی کنترل میزبانی است . اداره و کنترل این ها به تجربه OS است .

میزبانی خالی 2

نمایند 1- دید حقیقی مسافر از اطلاعات و راه اندازی بعد

اطلاعاتی که در آن جاهای مختلف ذخیره می شوند (روی دیسک ، CD ، بندهای دستی) Data می باشد
نه ، مسافر می بینند و اگر چیزی را می بینند این اطلاعات می باشد و می توانند در یکای قوی حقیقی خود کار را از این جا چیزی اضافه کنند و به هر دو اطلاعات چیزهای کلی و اطلاعاتی که در این کار می کنند که این کار را می بینند و این تفاوت ها را می بینند

Subject:

Year . Month . Date . ()

کتابخانه مدیریت فایل

ایجاد و حذف فایل

ایجاد و حذف راجه (directory)

کتابخانه‌های برای کار کردن با فایل

هفته پنجم: 7.9

سیستم مدیریت فایل ها Command Interpreter System

و اما بخشی از OS نیست چیزی است که هم با آن سروکار دارند

و سه راهی برای کار کردن به وسیله آن بتوانند فرمان دهد

سیستم عامل برای این امر از برای است که توسط از طریق System Call

ی بتواند با آن تماس گرفت یا هر چیزی را از آن تقاضا کرد

از همین درخواست های آنرا هم مستقیم طلب کنیم

مستقیم سیستم عامل درخواست های سیستمی است

پسندید به عنوان یک فرآیند

exit ()

که به OS می آید این فرآیند تا اینکه تمام مناسبت را از آن

آغاز به فرآیند (پسندید کردن یک فرآیند) create ()

که نیازمند سرویسها از ایند را مناسبت می دهد

malloc ()

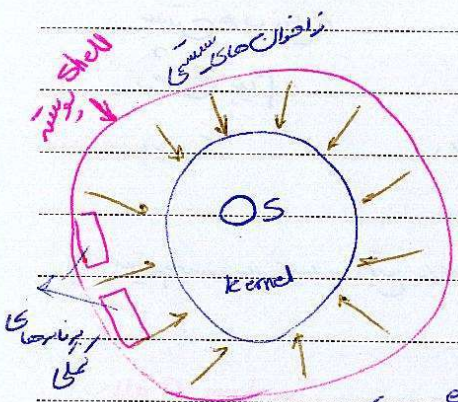
سیستم مدیریت فایل ها:

معمولاً یک سری برنامه های عملی که شامل یک بسته دور OS را می تواند عمل

نسیب در برای این امر از تعداد زیادی System Call استفاده می کند هر وقت که copy یک برنامه است

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

به کار خود ادامه می دهد



Subject:

Year. Month. Date. ()

copy F_1, F_2

مجموعی ایک آپٹی

سٹریٹجی میں تبدیلی

openFile ()

سیستم

ایک ایسی فرنی (مقصد F_2 تک پہنچانے کے لیے) → صورت وجود میں فرنی

→ صورت میں loop اطلاعات F_1 سے F_2 تک پہنچانے کے لیے

از بنا فرنی کے لیے دائرہ کار کو درست بنانے کے لیے

طریقہ: از طریق فرنی کے

فرنی کے

سیستم

فرنی

→ فرنی کے $error$ کا اسیٹھ کرنا، فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا

سیستم کے فرنی کے $error$ کو درست کرنا

System Calls

فرنی کے $error$ کو درست کرنا

Interrupt کی فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا

LD $R_1, Parm$

LD $R_2, Parm$

INT 28

فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا (push on stack)

فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا

فرنی کے $error$ کو درست کرنا

Subject :

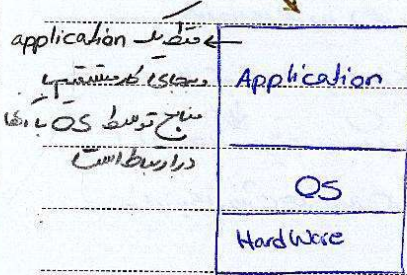
Year . Month . Date . ()

معمولی حالت در سیستم های جاوا برای mode برای سیستم مورد

پس از این

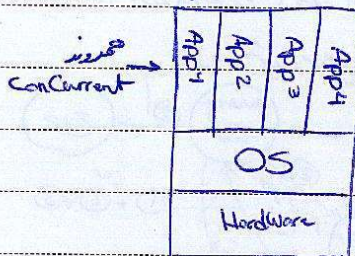
ماشین مجازی (Virtual Machine)

اول به سیستم ماشین رانق چیست



از برای اجرای OS استفاده می کنند تا بتوانند منابع استفاده کنند

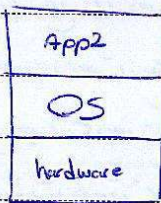
(3)



هر یک از application ها application ها هم بصورت همزمان در حال اجراست و کار هر کدام فقط در سطحی می باشد (همزمانه) و در اختیار دارد

اگر دیدیم برنامه ها در ماشین های یک تا سه که در حال اجراست و منابع را می بینیم و متوجه می شویم که در منابع با یکدیگر تداخل دارند این ... در واقع می تواند جای این است که OS یکی بعد از دیگری اجرا کند OS است

application 2



این واقعیت است که در واقعیت (3) است

و تداخل منابع از دید برنامه ها نمی است و چند OS

ماشین مجازی

وقتی که در هر یکی در اجرای برنامه ای در وقت برنامه های مختلف عمل می کنند در واقع است

منه این را می خوانند delay می بینند

Subject:

Year. Month. Date. ()

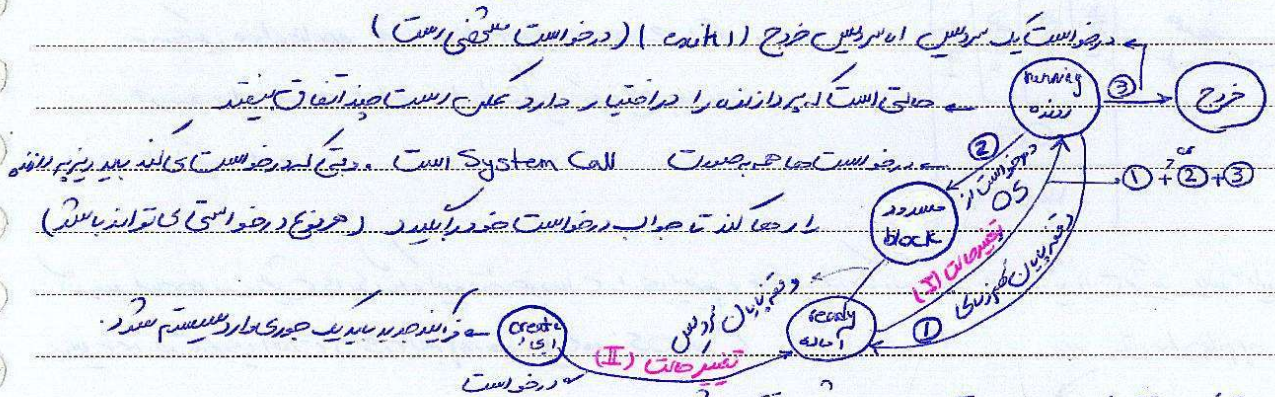
مدیریت فرآیند Process Management

تعریف فرآیند: برنامه در حال اجرا
فرآیندهای همزمان (همزمان)

زمان بندی (نوبت بندی) فرآیند: فرآیندها همزمان می توانند پردازنده را در اختیار بگیرند و برنامه باید یکی یکی پردازنده را در اختیار بگیرد که در این حالت سیستم نوبت بندی (Round Robin) خواهد داشت (نوبت بندی).
از ویژگی های زمان بندی: جزای این سیستم ها چگونه است و کنترل آنها باید چگونه باشد.

حالت فرآیند Process State

فرآیند در طول زندگی خودش حالات مختلفی دارد در این حالات اما تغییر می کند



در حالت خروج تمام منابع را از سیستم می برد و می کشد
وقتی در حالت معلق در است زمانی اتفاق می افتد که جواب درخواست آمده از سیستم (مورد تقاضای آنگونه) می تواند پردازنده را به این بخش چون دست نمی است پس باید در صف تردد تا حرمقت از بخش بگذرد پردازنده را در اختیار بگیرد پس به حالت آماده می رود تا حرمقت پردازنده را در اختیار گرفت دوباره running می شود یعنی به این حال هم حق دارد در آن پردازنده را در وسط کار این بگیرد و به پردازنده برگرداند
در تمام مراحل هم برای اجرای او باید به حالت آماده بود چون برای اجرای او باید در حافظه ای آماده باشد
را همین می داریم

← جان سرور که توسط رنده اعلام می شود

تغییر حالت ها به کار می رسد گرفته
که با این کار باید تغییر mode می تواند
که با این کار می توان

Subject:

Year . Month . Date . ()

دو تا تغییر حالت توضیح داده بشود و تغییر حالت (I)

(II) .. ✓

تغییر حالت (I)

زنی که حرکت از تغییر حالت صایه که باعث شود بر نامه ای که در running است این حالت را ترک کند برود از آنده را ترک کند
این تغییر حالت اتفاق می افتد یعنی هر گاه یکی از تغییر حالت های 1 یا 2 یا 3 اتفاق بیفتد

تغییر حالت (II)

ایجاد فرآیند در خواست است و فرآیند صایه زنی ای که فرآیند موجود در خواست ایجاد از آن بعد جواله ای در فرآیند
یک System Call است

(point) ایجاد فرآیند توسط سیستم در خواست یک فرآیند می شود (توسط یک فرآیند دیگر)

وقتی ما مثلا برای یک Icon، فایل یا فیلم فرآیند ایجاد می کنیم و ای که فرآیند اصلا توسط کاربر ایجاد می شود

(point) ایجاد فرآیند توسط فرآیند دیگر ایجاد می شود و راه دیگری نیست

التر create که توسط فرآیند ایجاد می شود پس فرآیند اولیه چطور می تواند ایجاد می شود ؟
کاربری کاره است ؟
ایمان

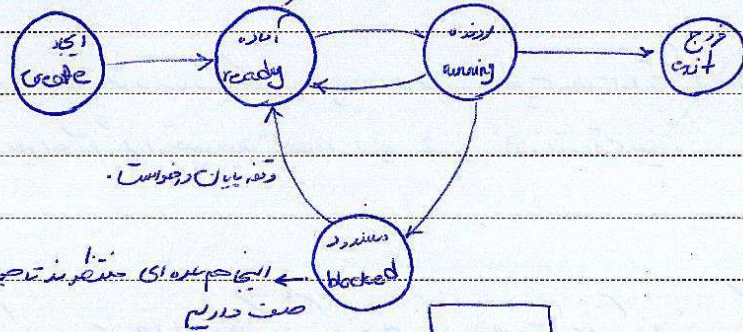
هنگام درخواستی با عیب می شود به نامه برود از آنده را از دست برده و در خواست create و این تغییر حالت ها
باعث از دست دادن پرونده می شود

Subject:

Year. Month. Date. ()

خالد حسینی 7, 19

ممكن ان تستفيد من دروسنا كما تستفيد من دروسنا كما تستفيد من دروسنا
من حيث دروسنا كمن يستفيد من دروسنا كما تستفيد من دروسنا



بعضی از تغییر حالتها ممکن است بر اساس حجم وقت و حجم درخواست باشد
تغییر حالتها بر اساس وقت و درخواست

سیستم عامل توسط وقت یا درخواست به اجرای افتد

ایجاد فرآیند توسط سیستم عامل به منظور اجرای برنامه است که یک برنامه که در حالت running است این درخواست را بکند

بر عنوان کاربری در سیستم برنامه های مختلفی اجرا می شود و چه چیزی؟

یک فرآیند داریم که دائما در حال اجرا است در این اجرا با monitor
Monitor فرآیند مانیتور در دزدی را چک می کند مثلا وقتی یک طلب در حال اجرای فرآیند جدید صورت می گیرد این ای دی الوند
کاهش حجم است و کار دیگر ای دی الوند فرآیند ای دی الوند را بر اساس دوری ها که می دهد این ای دی الوند
چون توسط فرآیند ایجاد فرآیند است پس اگر فرآیند در حالت blocked است و کار ای دی الوند
بنامه monitor وقتی که دوری آمد توسط OS بیرون کشیده می شود و بر بازنده در اختیارش قرار داده می شود

فرآیند به صورت فیزیکی چیست؟ برنامه در حال اجرای می باشد OS چگونه این فرآیند را اجرا می کند؟
وقتی برنامه به فرآیند تبدیل می شود OS باید منابعی را به آن بدهد پس در حال اجرای برنامه ای برای آن بسازد OS برای
هر که این ساختار داده ای بسازد و همین است که باعث می شود OS این ساختار بسازد این ساختار یکی ندارد که آن

Subject:

Year. Month. Date. ()

Process control Block (PCB) (مجموعه‌ای است از داده‌ها) که برای هر برنامه‌ی در حال اجرا وجود دارد تا بتوانیم
 است از هر OS PCB را برای فرآیند است. خروجی این PCB را برای ثبت و ثبتی حذف می‌کنیم
 حذف می‌شود و منابع آزادی می‌شوند.

داخل PCB چیست؟

Process state - حالت فرآیند

program counter - برنامه شماره
 برای اینکه می‌توانیم اجرای برنامه‌ی چه هست و وقتی اجرای برنامه‌ی چه می‌شود

CPU registers - رجیسترها
 برای اینکه می‌توانیم اجرای برنامه‌ی چه می‌شود و در CPU برادر اجرا می‌شود

stack pointer - اشاره‌گر به پشته

Memory management information - اطلاعات مدیریت حافظه

CPU scheduling Info - اطلاعات زمان بندی پردازنده

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری

load - بار

PCB - Process Control Block

Accounting Info - اطلاعات حسابداری