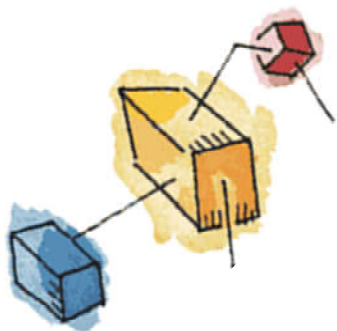


به نام خدا

فصل چهارم نخ ها، چند-پردازشی متقارن، و ریز هسته (بخش اول)

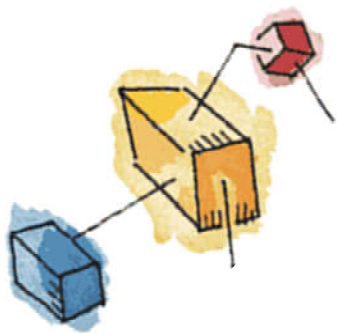
Threads, SMP, and Microkernels



سرفصل مطالب

- فرآیندها و نخ ها
- چند پردازی متقارن
- ریزهسته ها
- مدیریت نخ و چندپردازی متقارن در:
 - ویندوز
 - سولاریس
 - لینوکس

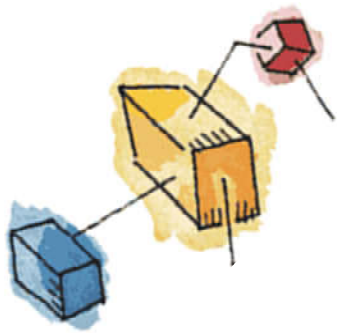




سرفصل مطالب

- فرآیندها و نخ ها
- چند پردازی متقارن
- ریزهسته ها
- مدیریت نخ و چندپردازی متقارن در:
 - ویندوز
 - سولاریس
 - لینوکس





فرآیندها و نخ ها

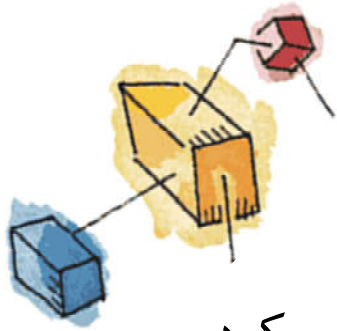
- فرآیند دو ویژگی دارد:

- تملک منابع: کنترل و تملک منابعی مثل حافظه، دستگاه های ورودی/خروجی، فایل ها و به فرآیند داده می شود.
- توزیع وقت پردازنده/ اجرا: فرآیند، موجودیتی است که توسط سیستم عامل مورد زمانبندی و توزیع وقت پردازنده قرار می گیرد.

- می توان این دو خصوصیت را از هم جدا کرد:

- به موجودیتی که مالکیت منابع به آن داده می شود، فرآیند یا وظیفه (task) گفته شود.
- به موجودیتی که وقت پردازنده به آن تخصیص داده می شود، نخ یا فرآیند سبک وزن گفته شود.

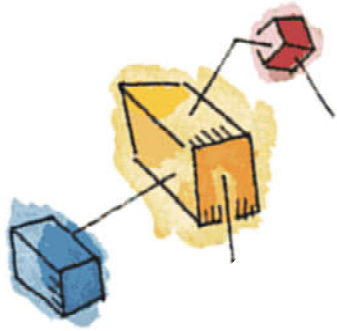




چند نخي (multi-threading)

- نخ کوچک ترین واحد کاری است که می توان وقت پردازنده را به آن توزیع کرد.
- چند نخي عبارت است از پشتیبانی سیستم عامل از چندین رشته یا نخ اجرای درون یک فرآیند واحد.
- سیستم عامل DOS تنها از یک نخ پشتیبانی می کرد.
- Unix از چندین فرآیند کاربر پشتیبانی نموده اما به ازای هر فرآیند تنها از یک نخ پشتیبانی می کند.
- اما Windows، Solaris، Linux و چندین سیستم عامل دیگر از چند نخي پشتیبانی می کنند.





چند نخه (multi-threading)

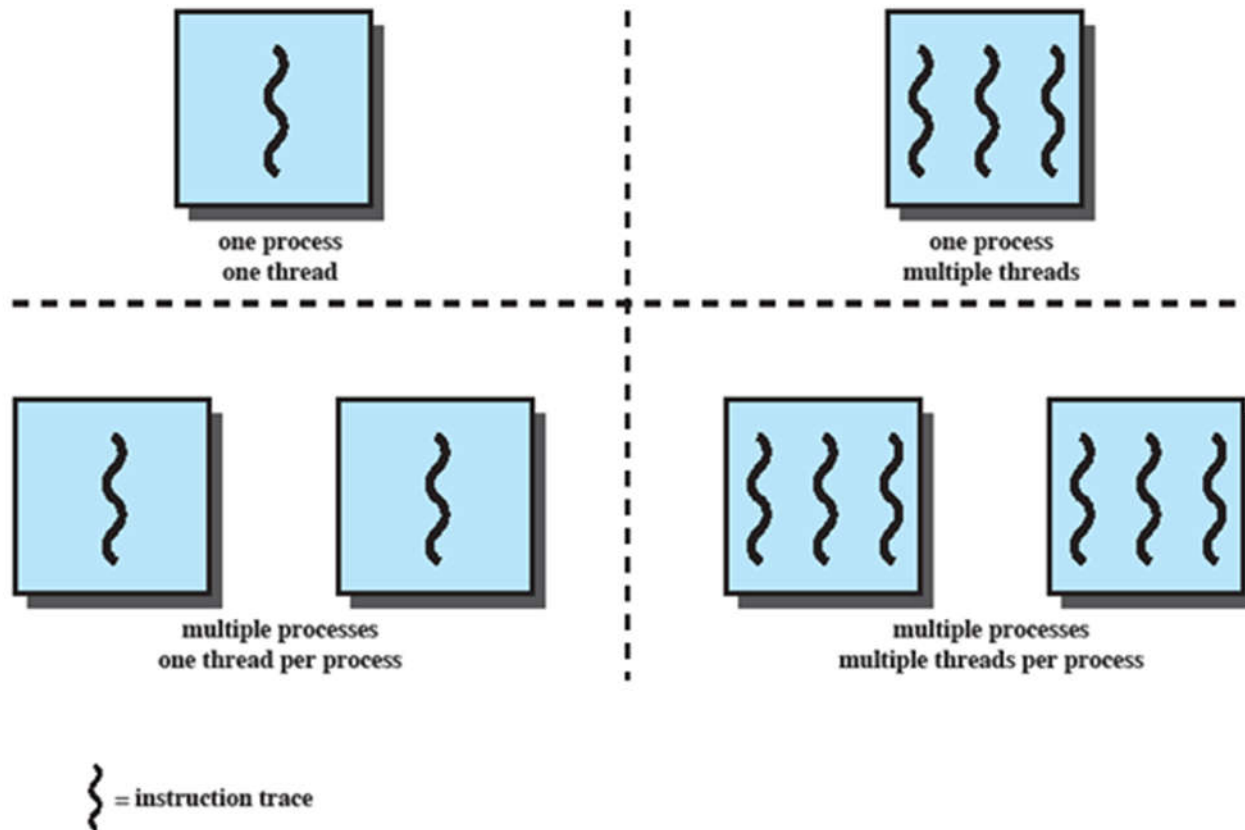
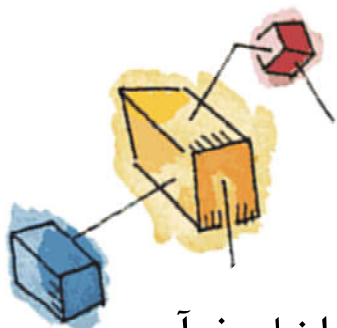


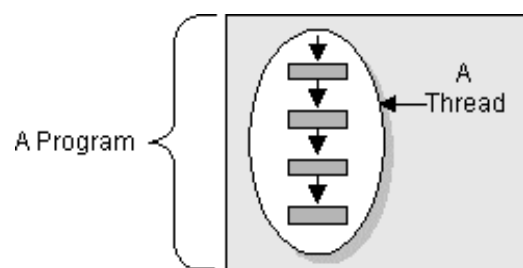
Figure 4.1 Threads and Processes [ANDE97]



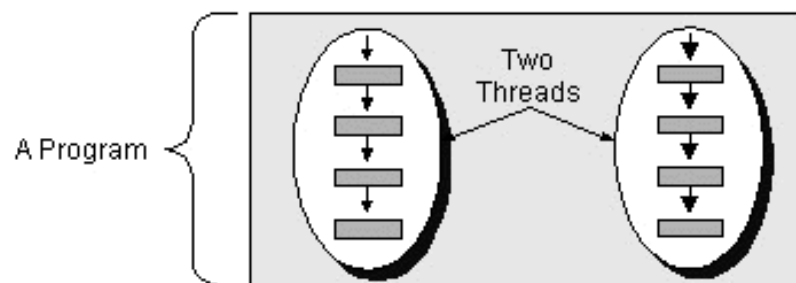


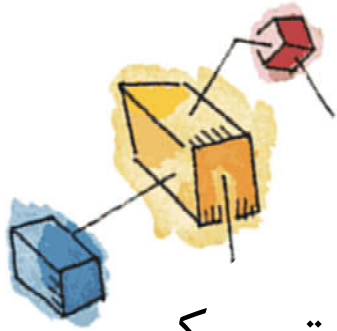
نخ و فرآیند

- یک نخ به خودی خود یک برنامه نیست و نمی تواند اجرا شود. بلکه در داخل فرآیند اجرایی یک برنامه اجرا می شود.



- ممکن است در یک زمان چندین نخ در یک برنامه در حال اجرا باشند و کارهای متفاوتی را انجام دهند.





اجزاء های مستقل و مشترک در نخ ها

- یک نخ از یک شناسه نخ، یک شمارنده برنامه، یک مجموعه ثبات و یک پشته تشکیل شده است.

- تقریباً همه چیز را با سایر نخ های همان فرآیند به اشتراک می گذارد.

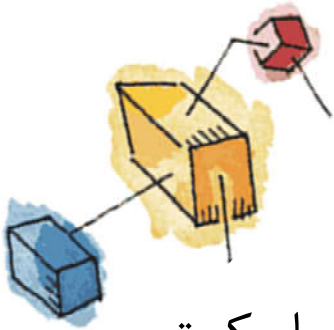
- بخش کد (اغلب)

- بخش داده

- منابع در اختیار گذاشته شده توسط سیستم عامل مانند پرونده های باز.



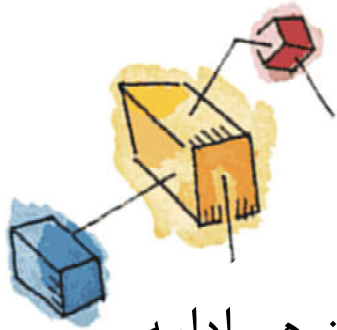
مزایای نخ ها



- ایجاد یک نخ جدید در یک فرآیند موجود از ایجاد یک فرآیند جدید بسیار کمتر وقت می برد.
- پایان دادن به یک نخ سریعتر است از پایان دادن به یک فرآیند.
- انتقال کنترل بین دو نخ داخل یک فرآیند (یعنی تعویض نخ) کمتر وقت می گیرد نسبت به تعویض بین دو فرآیند.
- از آنجا که نخ های داخل یک فرآیند از حافظه و فایل های مشترکی استفاده می کنند، می توانند بدون دخالت هسته با یکدیگر ارتباط برقرار کنند.



فواید برنامه نویسی چندنخی



- پاسخ دهی (Responsiveness)

– از آنجا که اجرای برنامه حتی در صورت متوقف شدن بخشی از آن باز هم ادامه می یابد، پاسخ دهی برنامه بهبود می یابد.

- اشتراک منابع (Resource Sharing)

– به صورت پیش فرض، نخ ها حافظه و منابع فرآیند را به اشتراک می گذارند.

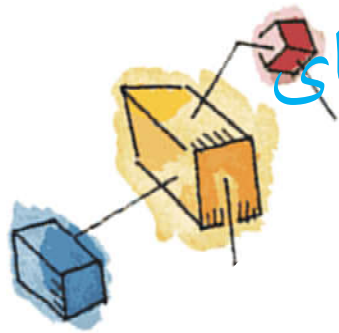
- صرفه جویی (Economy) در منابع سیستمی

– ایجاد یک نخ به مراتب ارزانتر از ایجاد یک فرآیند است.

- توسعه پذیری در معماری های چندپردازنده (Scalability)

– اجرای هر نخ روی یک پردازنده \Leftarrow افزایش درجه توازی و کارایی

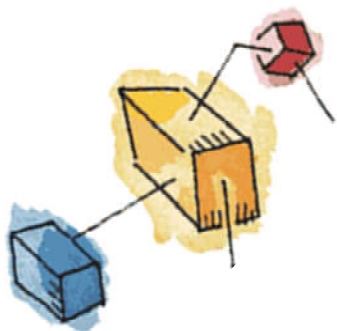




یک انگیزه برای برنامه نویسی چند نخی: سیستم های چندهسته ای (Multicore Programming)

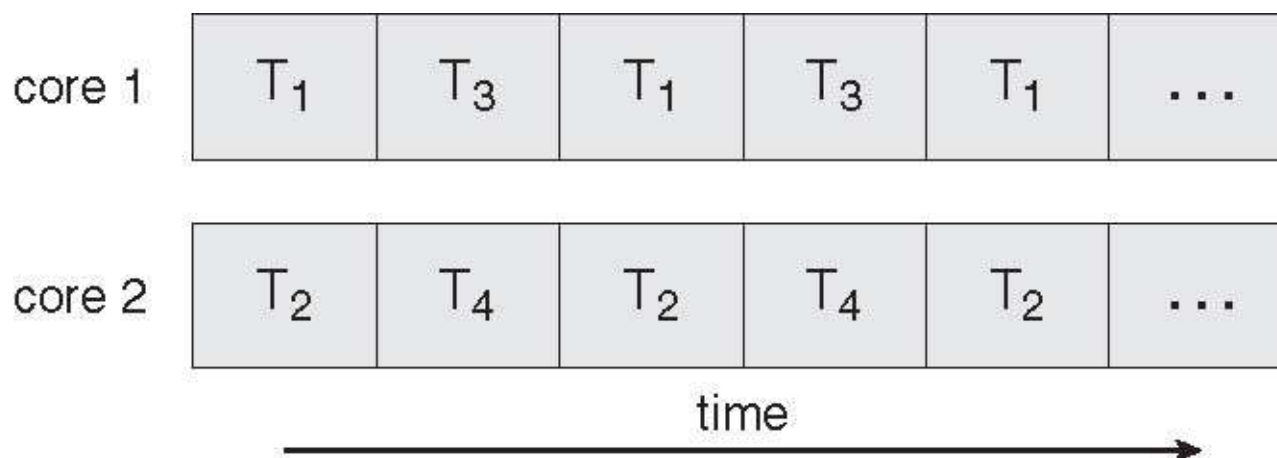
- سیستم چندهسته ای: قرار دادن چندین هسته محاسباتی در یک تراشه
- برنامه نویسی چندنخی راهکاری برای استفاده کارآمد از چندهسته را فراهم می کند.
- گرایش به سمت سیستم های چندهسته ای باعث شد که طراحان و برنامه نویسان سعی کنند از هسته های چندگانه استفاده بهتری کنند.
- طراحان سیستم عامل باید الگوریتم های زمانبندی بنویسند که از هسته های چندگانه استفاده بهتری بکنند.
- برنامه نویسان کاربردی نیز باید برنامه های خود را به صورت چندنخی بنویسند تا از سیستم های چندهسته ای استفاده بهینه کنند.



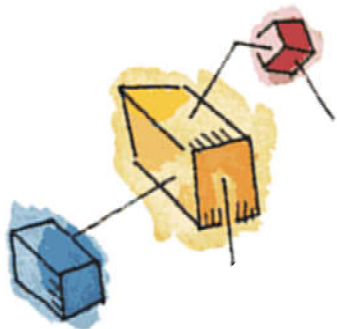


اجرای موازی در سیستم چند هسته ای

- در صورتی که سیستم چند هسته ای داشته باشیم
- اجرای برنامه کاربردی با چهار نخ:
- همزمانی به این صورت است که نخ ها می توانند به طور موازی اجرا شوند به طوری که سیستم می تواند به هر هسته نخ جداگانه ای را نسبت دهد.



مثال هایی از چند نخي

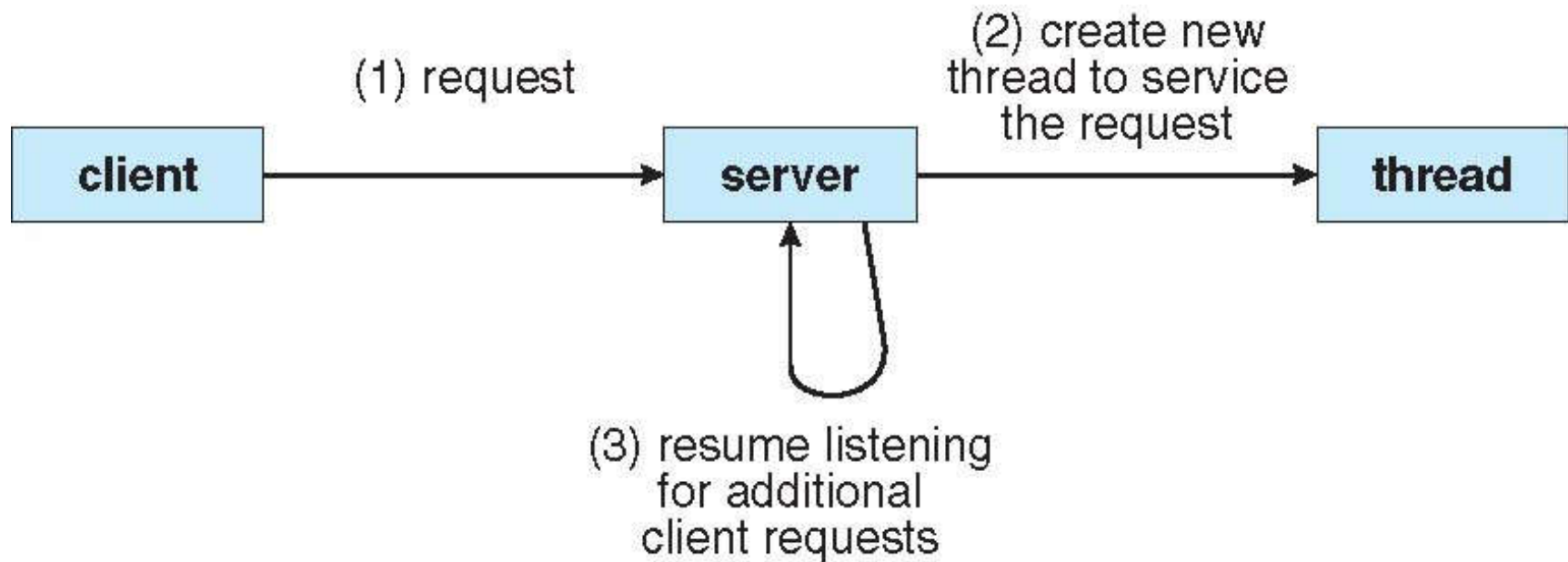
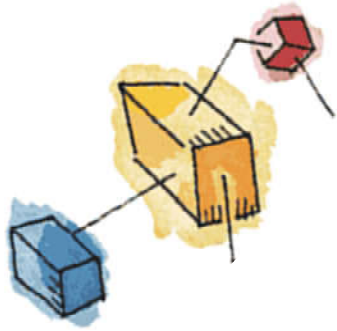


- بسیاری از پکیج های نرم افزاری چند نخي هستند:
 - مرورگر وب: نخي برای نمایش تصاویر یا متن، نخي برای بازیابی داده ها از شبکه
 - واژه پرداز (مثلا نرم افزار Word) : نخي برای نمایش گرافیکی، نخي برای پاسخ به کلیدهای فشرده شده توسط کاربر، نخي برای بررسی گرامر و املا، و ...
- در برخی موارد، ممکن است برنامه کاربردی چندین کار مشابه انجام دهد:
 - وب سرور: پاسخ دادن به درخواست های همزمان کلاینتها. دو رهیافت:
 - یک فرآیند برای پاسخ دادن به هر کلاینت
 - یک نخ برای پاسخ دادن به هر کلاینت

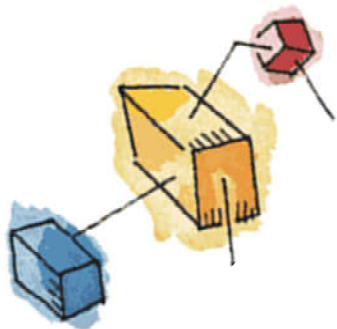
هسته بسیاری از سیستم عامل ها به صورت چند نخي پیاده سازی شده است.



مثال دیگر: معماری سرور چند نخه



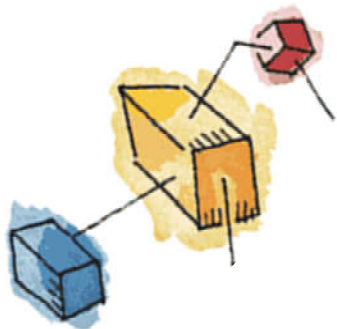
نخ و فرآیند



- بعضا به جای واژه «نخ» از عبارت «فرآیند سبک وزن» استفاده می شود.
- نخ یک فرآیند سبک وزن است، چرا که در متن یک فرآیند اجرا می شود و از منابع تخصیص یافته به آن فرآیند استفاده می کند.
- نخ نیز همچون فرآیند دارای یک کنترل ترتیبی ساده است و هر نخ شمارنده برنامه و پشته خاص خود را دارد.
- با تعلیق یک فرآیند، تمامی نخ های آن فرآیند نیز معلق می شوند. چرا که تمامی آنها دارای فضای آدرس واحدی (همان فضای آدرس فرآیند) می باشند.
- خاتمه یک فرآیند، خاتمه یافتن تمامی نخ های درون آن فرآیند را به دنبال دارد.

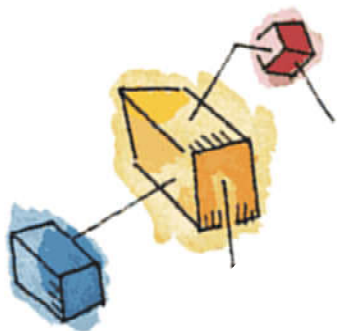


نخ و فرآیند

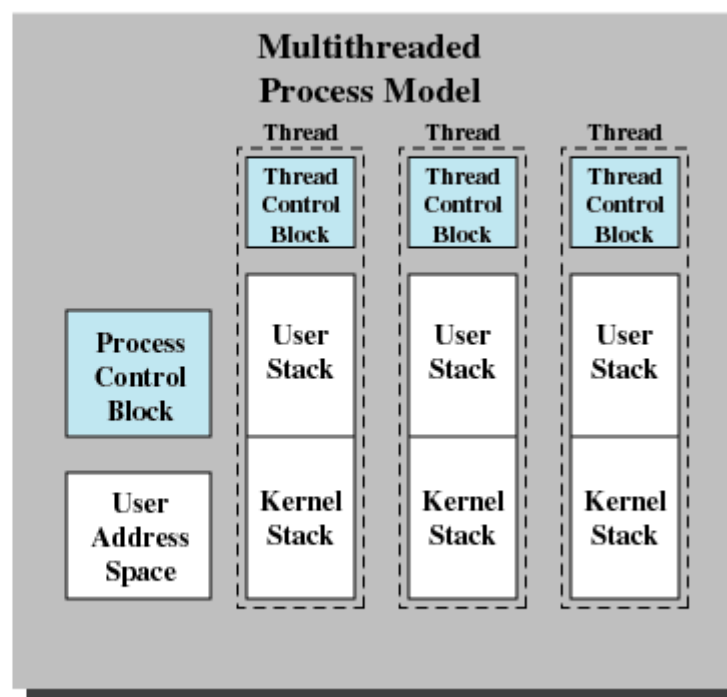
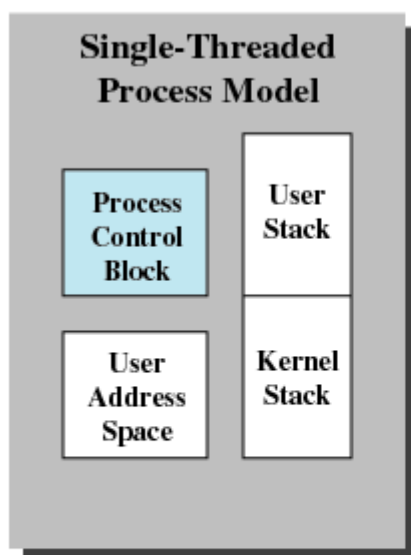


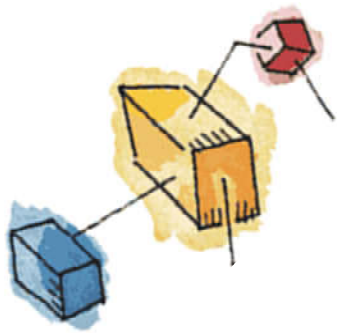
- در محیط چند-نخی موارد زیر همراه هر فرآیند هستند:
 - فضای آدرس مجازی که تصویر فرآیند را در بر دارد.
 - دسترسی حفاظت شده به پردازنده ها، فرآیندهای دیگر، فایل ها و منابع ورودی/خروجی
- در داخل هر فرآیند با یک یا چند نخ، هر یک از موارد زیر ممکن است وجود داشته باشد:
 - حالت اجرای نخ (اجرا، آماده، غیره)
 - متن نخ ذخیره شده
 - پشته اجرا
 - مقداری حافظه برای متغیرهای محلی هر نخ
 - دسترسی به حافظه و منابع فرآیند که مورد اشتراک تمام نخ های فرآیند هستند.





مدل فرآیند تک-نخی و چند-نخی

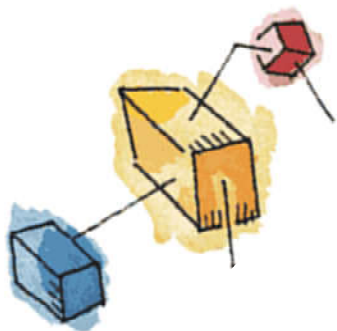




اعمالی که در سطح فرآیند انجام می شوند

- زمانبندی و توزیع وقت پردازنده بر مبنای نخ ها انجام می شود.
- با این حال، اعمالی وجود دارند که روی تمام نخ های یک فرآیند اثر می گذارند و سیستم عامل باید آنها را در سطح فرآیند مدیریت کند:
 - مثلاً معلق کردن یک فرآیند، شامل معلق شدن همه نخ های آن فرآیند است. چون همه نخ ها از فضای آدرس مشترکی استفاده می کنند.
 - با خاتمه یافتن یک فرآیند، همه نخ های آن فرآیند را خاتمه می دهد.



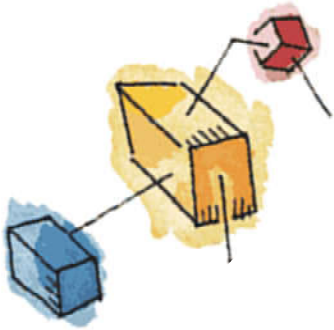


عملکرد نخ

- دو مبحث در مورد نخ عبارتند از:
 - حالات نخ ها
 - همگام سازی نخ ها (Synchronisation)
- که در ادامه بررسی می شوند.



حالات نخ ها



- نخ ها نیز مانند فرایندها دارای حالات اصلی اجرا، آماده، و مسدود هستند.
- حالت معلق برای نخ بی معنا است، چرا که تعلیق در سطح فرآیند انجام می شود.

- چهار عمل نخ ها، باعث تغییر حالت نخ می شود:

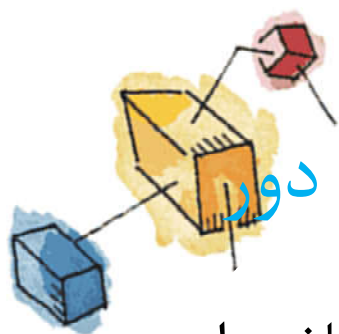
– ایجاد یا زایش: هنگامی که یک فرآیند ایجاد می شود، خود به خود یک نخ در آن ایجاد می شود. علاوه بر آن، ممکن است یک نخ در داخل یک فرآیند، نخ دیگری را ایجاد کند.

– مسدود شدن: وقتی لازم شود نخ برای بروز حادثه ای منتظر بماند، مسدود می شود. ممکن است پردازنده به اجرای نخ دیگری از آن فرآیند یا فرآیند دیگر پردازد.

– رفع مسدود بودن: وقتی حادثه مورد نظر رخ دهد، نخ از حالت مسدود خارج می شود.

– پایان: آزاد کردن متن ثبات ها و پشته های اجرایی نخ

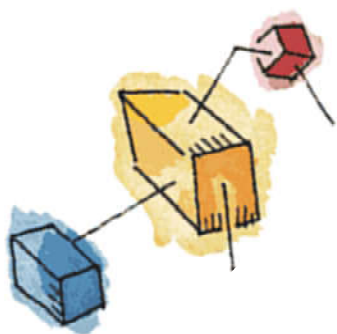




مثالی از تغییر حالت در نخ ها :فراخوانی رویه از راه دور

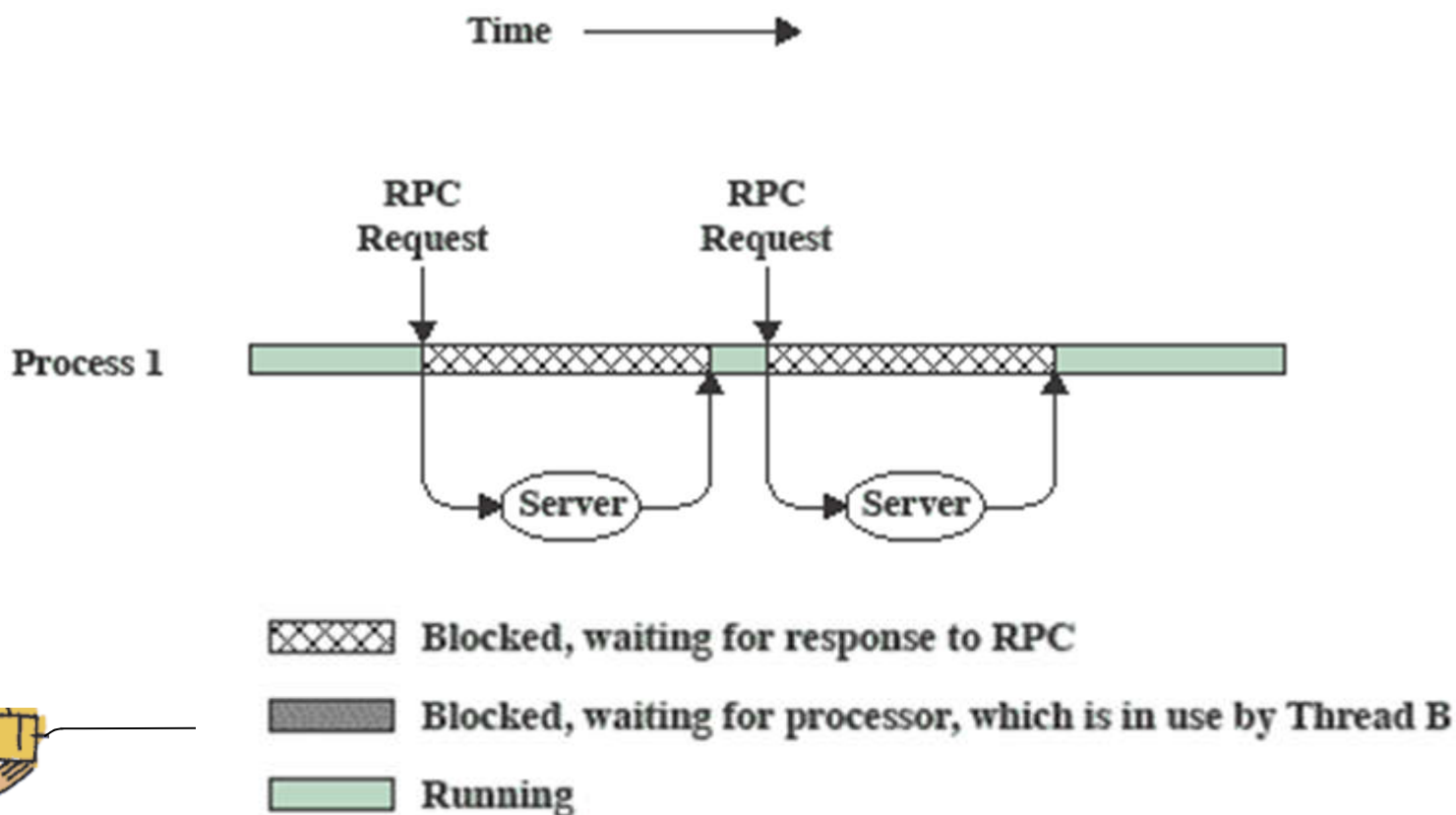
- برنامه ای دو فراخوانی رویه از راه دور (RPC) از دو میزبان مختلف را انجام می دهد.
- قرار است نتیجه این دو فراخوانی با یکدیگر ادغام شود و یک نتیجه ترکیبی به دست آید.

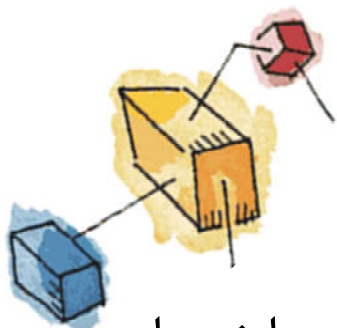




فراخوانی رویه از راه دور با استفاده از یک نخ واحد

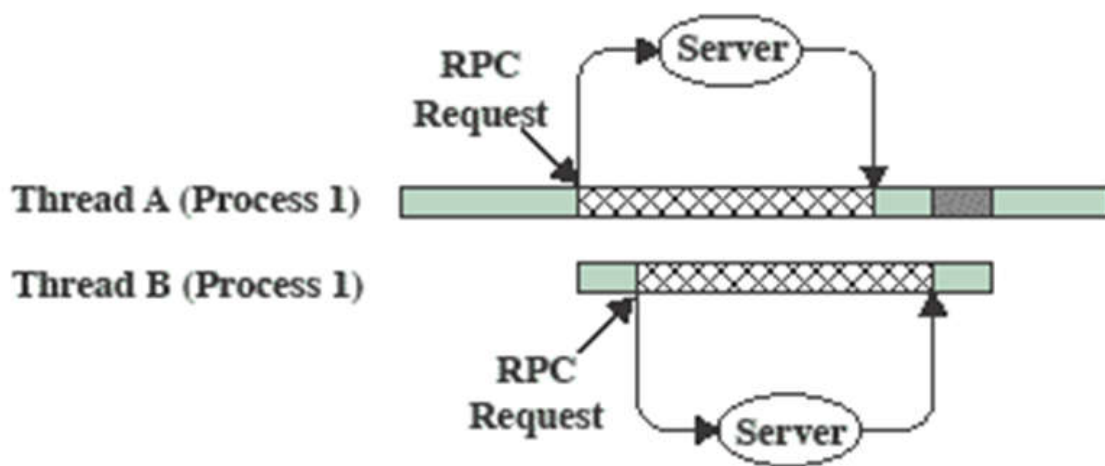
- در یک برنامه تک نخی نتایج به ترتیب به دست می آیند بنابراین برنامه به نوبت باید برای هر سرور صبر کند.








فراخوانی رویه از راه دور با استفاده از چند نخ

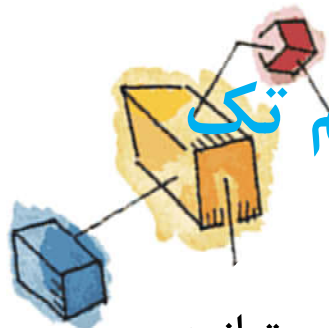
- استفاده از دو نخ باعث افزایش سرعت می شود. اگر از سیستم تک پردازنده ای استفاده شود، درخواست ها به ترتیب مطرح می شوند ولی انتظارها برای پاسخ ها به صورت همزمان است.



(b) RPC Using One Thread per Server (on a uniprocessor)

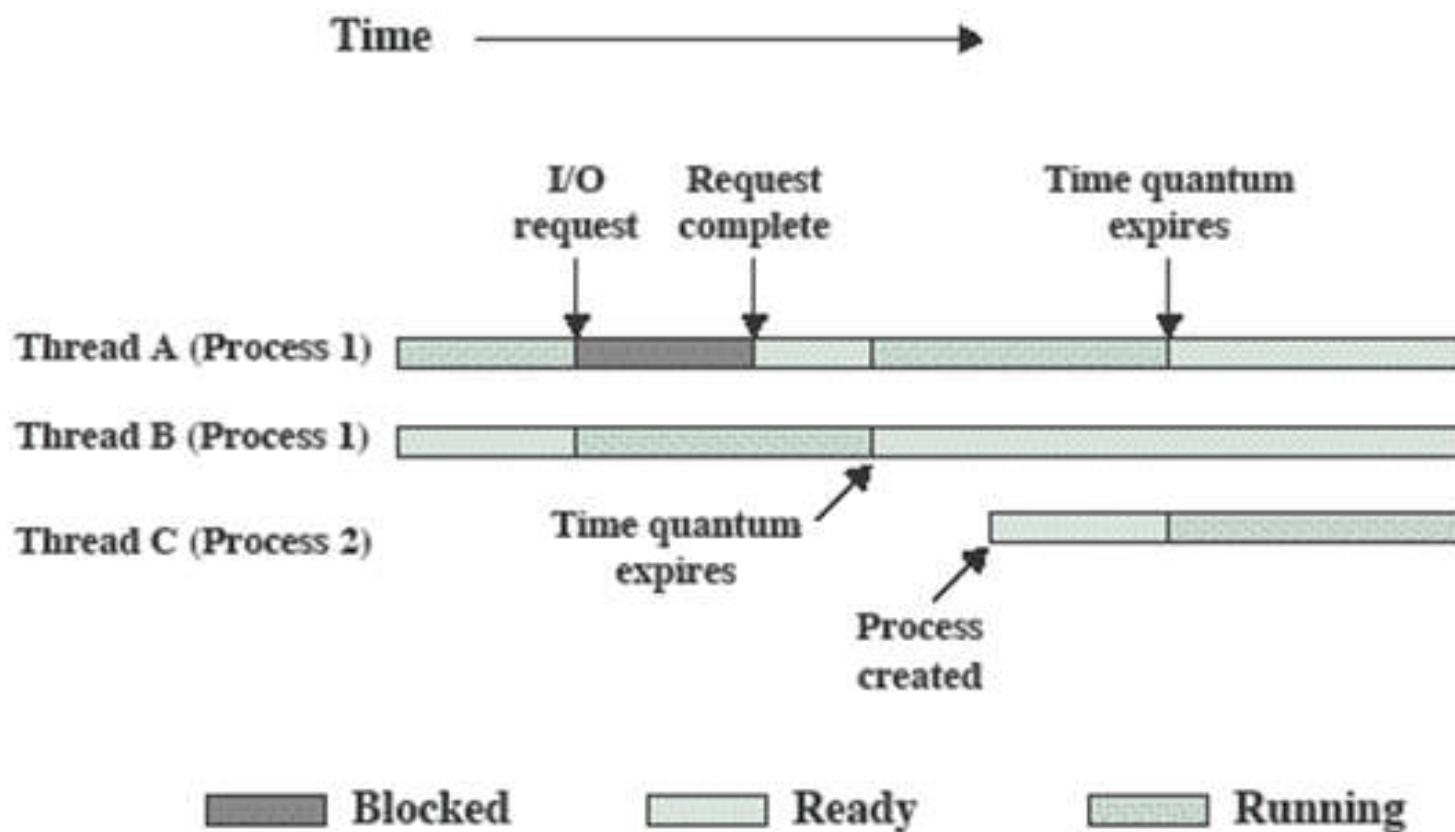
-  Blocked, waiting for response to RPC
-  Blocked, waiting for processor, which is in use by Thread B
-  Running

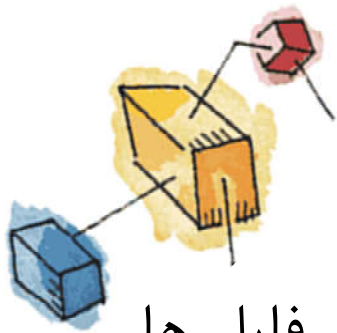




مثالی از چند-نخی (Multithreading) روی یک سیستم تک پردازنده

- در یک سیستم تک پردازنده ای چند برنامه ای، چند نخ از چند فرآیند می توانند در بین هم قرار بگیرند. هرگاه نخ در حال اجرا، مسدود شود و یا برهه زمانی آن به پایان برسد، پردازنده به نخ دیگری سپرده می شود.



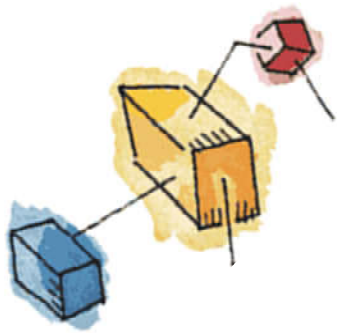


همگام سازی نخ ها

- تمام نخ های یک فرآیند در فضای آدرس و در منابع دیگر، مثل فایل ها شریک هستند.
- از این رو تغییر هر یک از منابع توسط یک نخ روی محیط نخ های دیگر آن فرآیند تاثیر می گذارد.
- بنابراین همگام سازی فعالیت های مختلف نخ ها ضروری است
- به گونه ای که در کار یکدیگر مداخله نکرده و ساختمان داده های یکدیگر را تخریب نکنند.
- مثلاً اگر هر دو قصد اضافه کردن عنصری به لیست پیوندی را داشته باشند ممکن است به شکل مطلوب پایان نیابد.

این موضوع در فصل های ۵ و ۶ بررسی خواهد شد.





دو دسته کلی برای پیاده سازی نخ ها

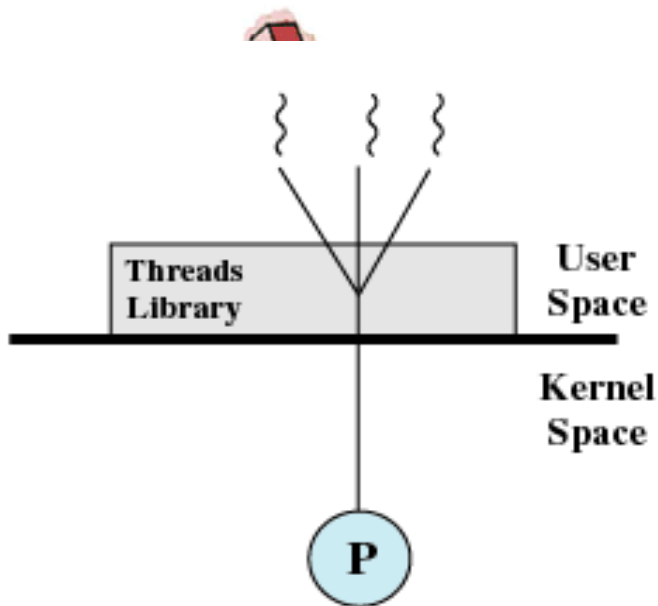
- دو دسته کلی برای پیاده سازی نخ ها وجود دارند:

- نخ های سطح کاربر یا User Level Thread (ULT)

- نخ های سطح هسته یا Kernel level Thread (KLT) که نخ های مورد حمایت هسته و یا فرآیندهای سبک وزن هم نامیده می شوند.



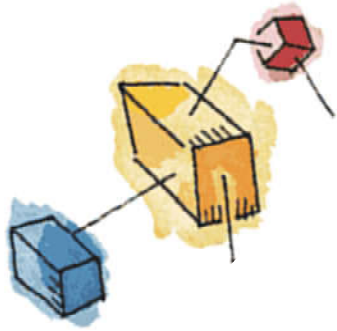
نخ های سطح کاربر



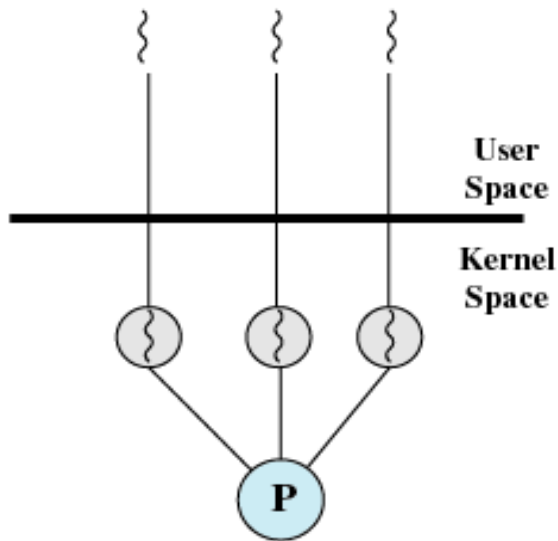
- هسته از وجود نخ ها بی اطلاع است.
- تمامی کار مدیریت نخ ها درون یک فرآیند واحد و توسط برنامه کاربردی انجام می شود.
- با استفاده از کتابخانه نخ ها که مجموعه ای از روال ها برای مدیریت نخ های سطح کاربر است، می توان برنامه ها را به صورت چند-نخی برنامه سازی کرد.
- کتابخانه نخ ها شامل کد ایجاد و تخریب نخ ها، کد انتقال پیام ها و داده ها بین نخ ها، کد زمانبندی اجرای نخ ها، و کد ذخیره و بار کردن متن نخ ها است.



نخ های سطح هسته



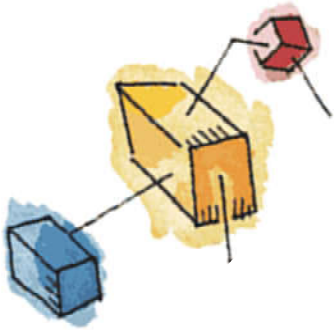
- تمام کار مدیریت نخ توسط هسته انجام می شود.
- زمانبندی بر مبنای نخ صورت می گیرد.
- هسته متن اطلاعات را برای فرآیندها و نخ ها نگهداری می کند. مدیریت نخ ها توسط برنامه کاربردی صورت نمی گیرد.
- Linux و ویندوز نمونه هایی از این رویکرد هستند.



(b) Pure kernel-level



مزایا و معایب نخ های سطح کاربر



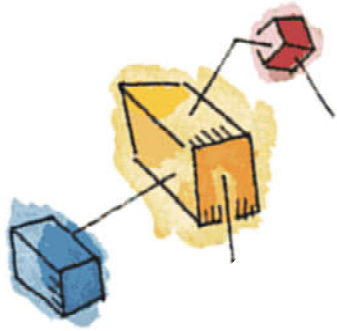
• معایب:

- زمانی که یک نخ یک فراخوانی سیستم را اجرا کند نه تنها آن نخ بلکه تمام نخ های آن فرآیند مسدود می شوند.
- یک کاربرد چند-نخی نمی تواند از امتیازات چندپردازشی بین نخ های خود استفاده نماید.
- هسته در هر زمان یک فرآیند را فقط به یک پردازنده نسبت می دهد. بنابراین در هر لحظه فقط یک نخ داخل فرآیند می تواند اجرا شود.

• مزایا:

- برای تعویض نخ نیازی به دخالت هسته نیست.
- می توان الگوریتم زمانبندی نخ ها را متناسب با کاربرد در نظر گرفت، بدون اینکه باری برای سیستم عامل داشته باشد.
- نخ های سطح کاربر می توانند بر روی هر سیستم عاملی اجرا شوند. هیچ تغییری در هسته برای حمایت از نخ های سطح کاربر لازم نیست.





مزایا و معایب نخ های سطح هسته

• مزایا:

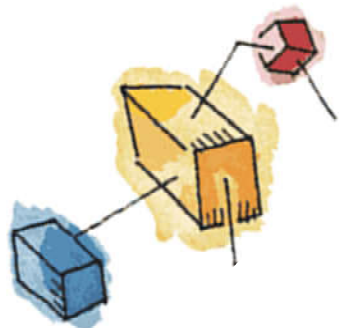
- هسته می تواند به طور همزمان نخ های چندگانه یک فرآیند واحد را روی پردازنده های متعدد زمانبندی نماید.
- اگر نخي از یک فرآیند مسدود شود، هسته می تواند نخ دیگری از همان فرآیند را زمانبندی نماید.
- خود روال های هسته نیز می توانند چند-نخی باشند.

• معایب:

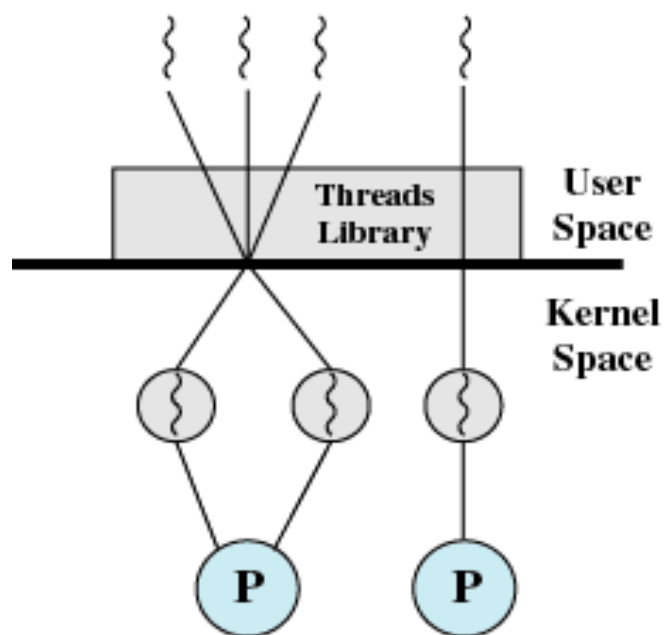
- انتقال کنترل از یک نخ به نخ دیگر در داخل یک فرآیند واحد، نیازمند تغییر حالت پردازنده به حالت هسته است.



راهکارهای ترکیبی



- بعضی از سیستم های عامل، ترکیبی از نخ های سطح کاربر و سطح هسته را ارائه می دهند.



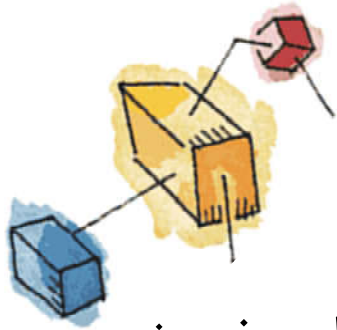
- تولید نخ به طور کامل در فضای کاربر در داخل یک برنامه کاربردی صورت می گیرد.

- قسمت عمده ی زمانبندی و همگام سازی نخ ها درون برنامه کاربردی انجام می شود.

- Solaris نمونه ای از این نوع سیستم عامل است.

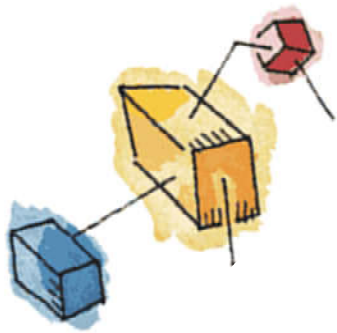


رابطه بین نخ ها و فرآیند ها



- **یک به یک:** هر نخ اجرا، یک فرآیند یکتا با فضای آدرس و منابع خویش است.
- **چند به یک:** یک فرآیند، فضای آدرس و مالکیت پویای منابع را تعریف می کند و ممکن است نخ های متعددی داخل آن فرآیند ایجاد و اجرا گردد.
- **یک به چند:** ممکن است یک نخ از محیط یک فرآیند به محیط فرآیند دیگر مهاجرت کند (حرکت نخ بین سیستم های مجزا).
- **چند به چند:** ترکیب خصوصیات موارد چند به یک و یک به چند.

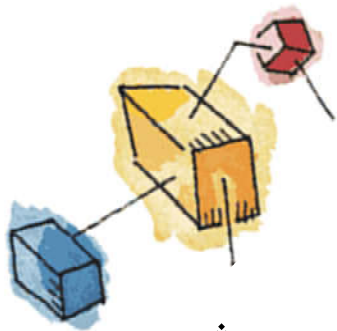




سرفصل مطالب

- فرآیندها و نخ ها
- چند پردازی متقارن
- ریزهسته ها
- مدیریت نخ و چندپردازی متقارن در:
 - ویندوز
 - سولاریس
 - لینوکس





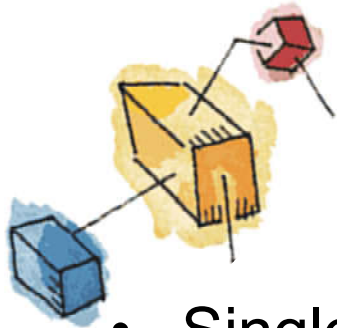
چندپردازنده ای متقارن

- به طور سنتی کامپیوتر به عنوان یک ماشین ترتیبی (ردیفی) دیده شده است.
- پردازنده با اجرای ردیفی و یکی یکی دستورالعمل های ماشین، برنامه را اجرا می کند.
- هر دستورالعمل دنباله ای از عمل ها است (واکشی دستورالعمل، واکشی عملوندها، انجام عمل و ذخیره نتایج).
- دو رویکرد متداول برای افزایش موازی سازی:

✓ چندپردازنده ای متقارن (Symmetric MultiProcessor (SMP

✓ خوشه ها (clusters)





دسته بندی سیستم های کامپیوتری

- Single Instruction Single Data (SISD)

- یک دستورالعمل یک داده

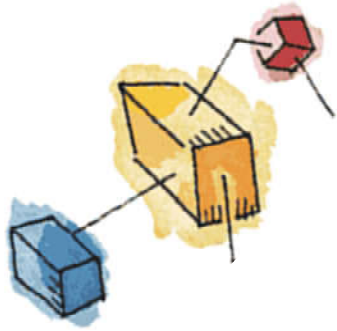
– یک پردازنده جریان واحدی از دستورالعمل را برای اعمال روی داده هایی که در یک واحد حافظه قرار دارند اجرا می کند.

- Multiple Instruction Single Data (MISD)

- چندین دستورالعمل یک داده

– یک دنباله واحد از داده ها به یک مجموعه از پردازنده ها منتقل شده و هر پردازنده دنباله ای متفاوت از دستورها را بر روی آن اجرا می کند. این ساختار هیچگاه پیاده سازی نشده است.





دسته بندی سیستم های کامپیوتری

- Single Instruction Multiple Data (SIMD)

- یک دستورالعمل چندین داده

– هر دستورالعمل توسط پردازنده های مختلف بر روی مجموعه داده های متفاوتی اجرا می شود.

- Multiple Instruction Multiple Data (MIMD)

- چندین دستورالعمل چندین داده

– مجموعه ای از پردازنده ها به طور همزمان دنباله های متفاوتی از دستورالعمل ها را روی مجموعه های متفاوتی از داده ها اجرا می کنند.



معماری پردازنده های موازی

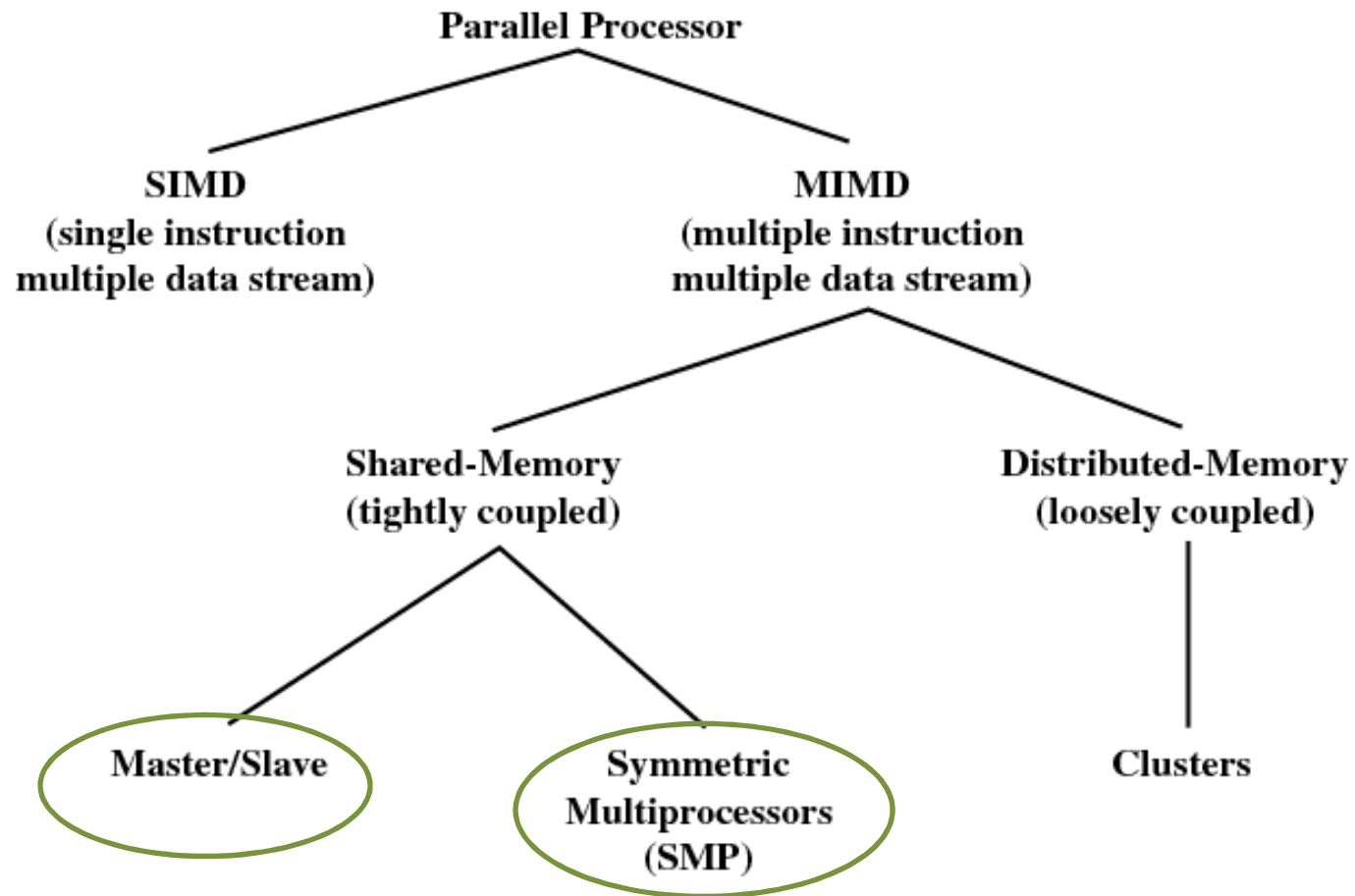
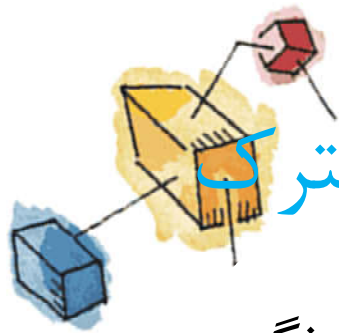


Figure 4.8 Parallel Processor Architectures



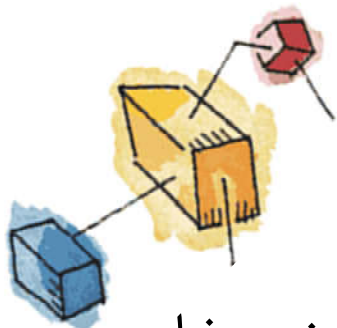
سیستم های چند پردازنده ای با حافظه مشترک

- سیستم های چند پردازنده ای با حافظه مشترک، بر اساس چگونگی تخصیص فرآیندها به پردازنده ها دو رویکرد دارند:

– رویکرد رئیس و مرئوس (master/slave)

– رویکرد چند-پردازشی متقارن (Symmetric Multi-Processing) یا به اختصار SMP





رویکرد رئیس و مرئوس

- در معماری رئیس-مرئوس، سیستم عامل همواره روی یک پردازنده خاص اجرا می شود.

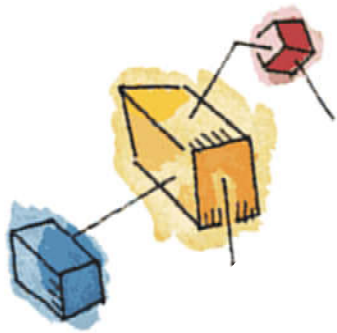
- پردازنده های دیگر فقط می توانند برنامه های کاربران و احتمالا برنامه های سودمند سیستم عامل را اجرا کنند.

- مسئولیت زمانبندی نخ ها و فرآیندها بر عهده رئیس است.

- اگر مرئوس نیاز به خدمتی داشته باشد باید درخواستش را به رئیس بفرستد.

- این رویکرد ساده است. باتوجه به اینکه پردازنده کنترل تمام حافظه و منابع ورودی/خروجی را در اختیار دارد، رفع درگیری ها ساده شده است.



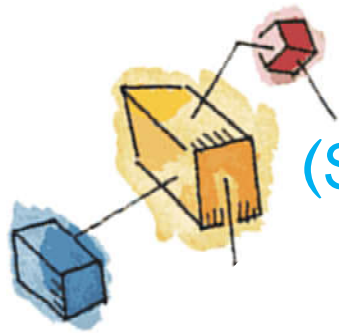


ضعف رویکرد رئیس و مرئوس

• ضعف ها:

- با خرابی رئیس تمام سیستم از کار می افتد.
- خود رئیس ممکن است گلوگاهی برای کارایی شود زیرا باید به تنهایی تمام زمانبندی ها و مدیریت فرآیندها را انجام دهد.





چند-پردازشی متقارن (Symmetric Multi-Processing)

- هسته می تواند بر روی هر پردازنده ای اجرا شود.
- معمولا هر پردازنده از بین نخ ها و فرآیندهای موجود، تعدادی را برای خود زمانبندی می کند.
- هسته می تواند به صورت فرآیندهای چندگانه با نخ های چندگانه ساخته شود.
- طراحی SMP مشکل است و موضوعات سازمان فیزیکی، ساختارهای اتصال داخلی، ارتباط بین پردازنده ها و سیستم عامل و ... را در بر دارد.



چندپردازنده ای متقارن

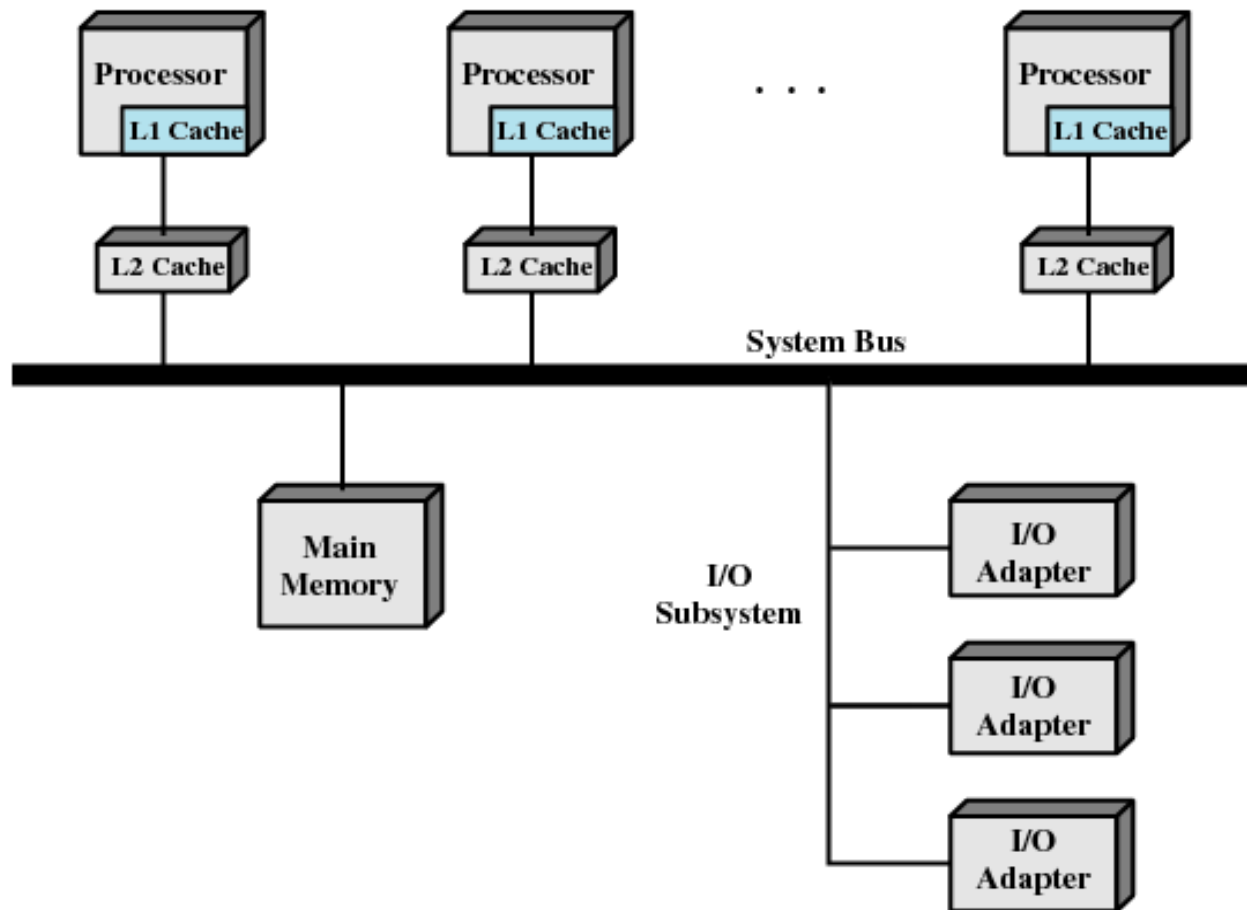
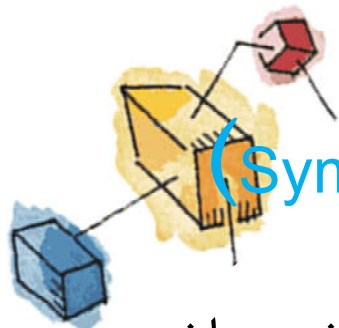


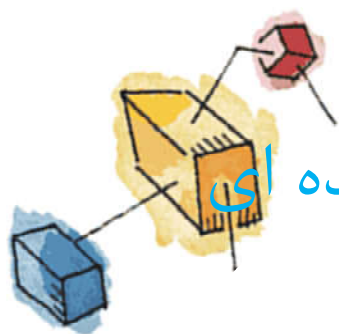
Figure 4.9 Symmetric Multiprocessor Organization



چند-پردازشی متقارن (Symmetric Multi-Processing)

- پردازنده ها می توانند از طریق حافظه مشترک با هم ارتباط داشته باشند (برای این منظور، پیام ها و اطلاعات در فضای آدرس مشترکی گذاشته می شود).
- امکان تبادل مستقیم بین پردازنده ها نیز وجود دارد.
- هر یک از پردازنده ها حداقل یک سطح از حافظه پنهان (کش) مخصوص به خود را دارد.
- استفاده از این حافظه پنهان، ملاحظات جدیدی را ایجاد می کند.
 - هر حافظه پنهان حاوی تصویری از بخشی از حافظه است.
 - تغییر کلمه ای از یک حافظه پنهان، ممکن است کلمه ای از حافظه پنهان پردازنده دیگر را بی اعتبار کند.





ملاحظات مهم در طراحی سیستم عامل های چند-پردازنده ای

- همزمانی نخ ها یا فرآیندها
- زمانبندی
- همگام سازی
- مدیریت حافظه
- قابلیت اطمینان و تحمل خطا

