

فصل چهارم

نخها، چندپردازی متقارن و ریز هسته ها

مباحث این فصل:

- فرایندها و نخها
- چند پردازشی متقارن
- ریز هسته ها

فرآیندها

فرآیند دو خصوصیت زیر را دارد:

- تملک منبع: (فرآیند یا وظیفه)

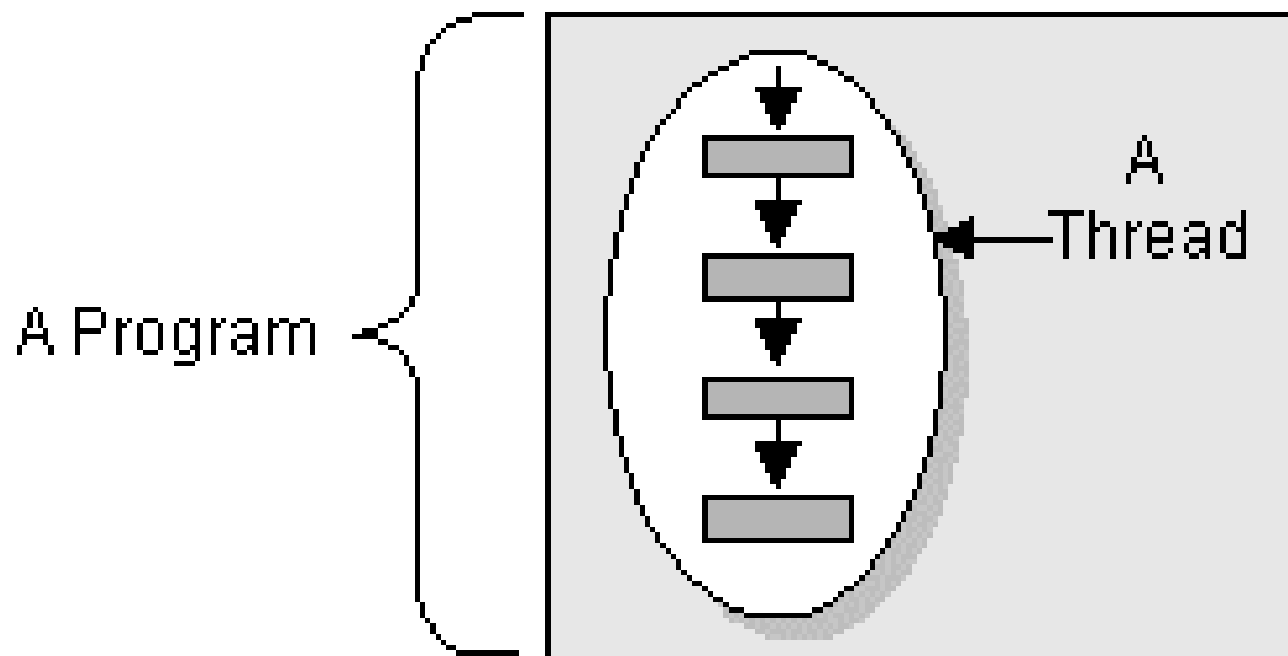
به هر فرآیند یک آدرس مجازی برای نگهداری تصویر فرآیند می دهد.

- توزیع وقت فرآیند: (نخ یا فرآیند سبک وزن)

فرآیند مسیر اجرای یک یا چند برنامه است و توسط سیستم عامل مورد زمان بندی و توزیع قرار می گیرد.

نخها

- شکل زیر روابط بین نخ و فرایند را نشان میدهد.



نخ چیست؟

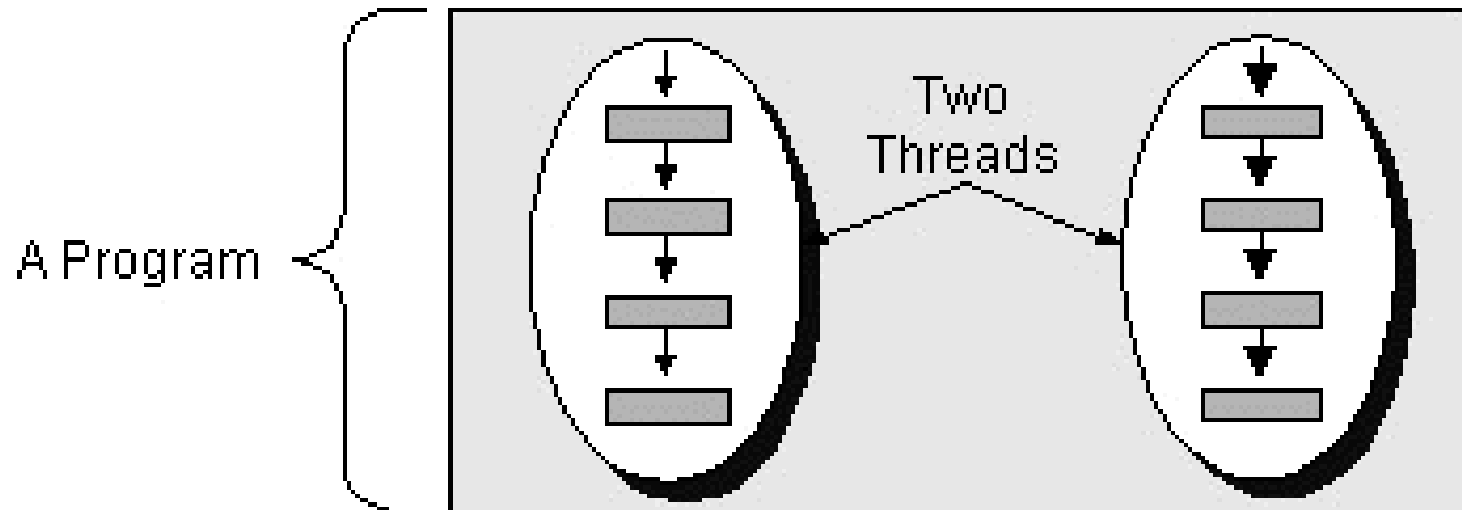
- نخها همانند برنامه نویسی ترتیبی دارای موارد زیرند.
 - نقطه شروع
 - ترتیب اجرا
 - نقطه پایانی
- در حین اجرای نخها در هر لحظه از زمان یک نقطه اجرای یکتا وجود دارد.
- یک نخ به خودی خود یک برنامه نیست و نمی تواند اجرا شود. بلکه در داخل برنامه اجرا میشود.

نخها و فرایندهای سبک وزن

- بعضی از متون به جای واژه نخ از واژه فرایند سبک وزن استفاده میکنند.
- یک نخ همانند یک فرایند واقعی است و در هر دو یک کنترل ترتیبی ساده وجود دارد. با این حال نخ یک فرایند سبک وزن است، چرا که در متن یک فرایند کامل اجرا میشود و از منابع تخصیص یافته به فرایند و محیط آن استفاده میکند.
- گاهی به نخ رد دستورالعمل نیز می گویند.

نخهای چندگانه:

- ممکن است در یک زمان چندین نخ در یک برنامه در حال اجرا باشند و کارهای متفاوتی را انجام دهند. شکل زیر این مطلب را نشان میدهد.



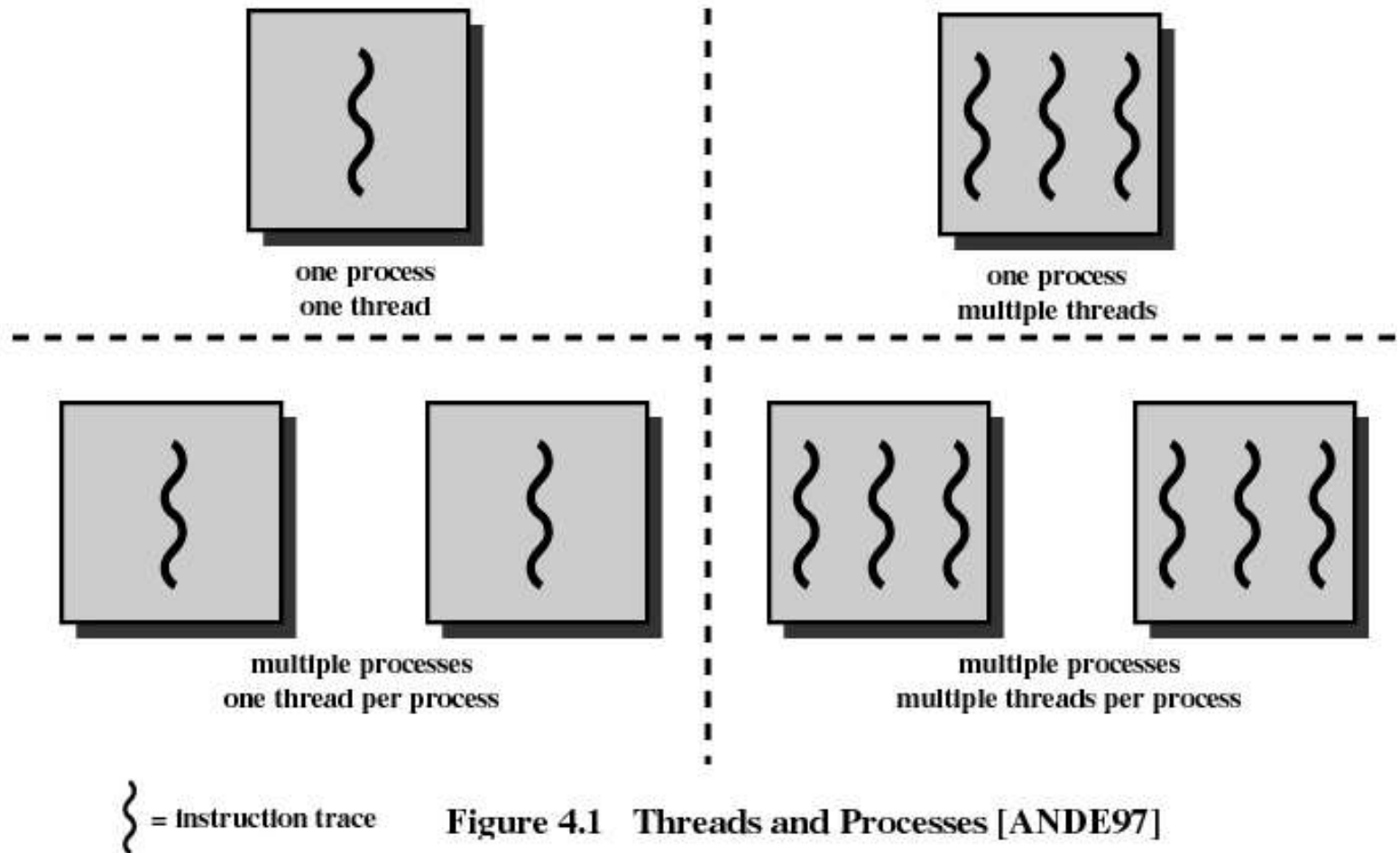
نخهای چندگانه:

- محیط زمان اجرای جاوا نمونه سیستمی از یک فرایند و چند نخ می باشد.
- نخهای متعدد میتوانند در یک فرایند، دو یا چند کار را همزمان با هم انجام میدهند.
- در چنین سیستمی شما میتوانید در حین بار شدن صفحه یا یک تصویر یا یک اپلت، یک صوت یا انیمیشن را به طور همزمان پخش کنید.
- به دلیل کنترل جریان ترتیبی نخها، هر نخ باید اطلاعات، منابع و متن مخصوص به خود را در حین اجرا داشته باشد.

چندنخی:

- چند نخی به قابلیت از سیستم عامل گفته میشود که از وجود چند نخ اجرا در یک فرایند واحد حمایت میکند.
- MS-DOS از فرایند تک کاربره و از نخ واحد حمایت میکند.
- محیط زمان اجرای جاوا نمونه سیستمی از یک فرایند و چند نخ می باشد.
- یونیکس از فرایندهای کاربران متعدد ولی فقط با یک نخ در هر فرایند حمایت میکند.
- ویندوز ۲۰۰۰، لینوکس، .. از فرایندهای چندنخی حمایت میکنند.

چندنخی و چند فرایندی:



فرایند در محیط چند نخی:

- در محیط چند نخی موارد زیر همراه هر فرایند هستند:
- فضای آدرس مجازی که تصویر فرایند را در بر دارد
- دسترسی حفاظت شده به پردازنده ها، فرایندهای دیگر، پرونده ها و منابع ورودی / خروجی

ساختمان نخ:

- در داخل هر فرایند یک یا چند نخ هر یک با موارد زیر ممکن است وجود داشته باشد:
- حالت اجرای نخ (اجرا، آماده، غیره)
- متن نخ ذخیره شده (که در حال اجرا نیست)
- پشته اجرا
- مقداری حافظه ایستا برای متغیرهای محلی هر نخ
- دسترسی به حافظه و منابع فرایند که مورد اشتراک تمام نخهای فرایند هستند.

مدل تک نخه و چند نخه:

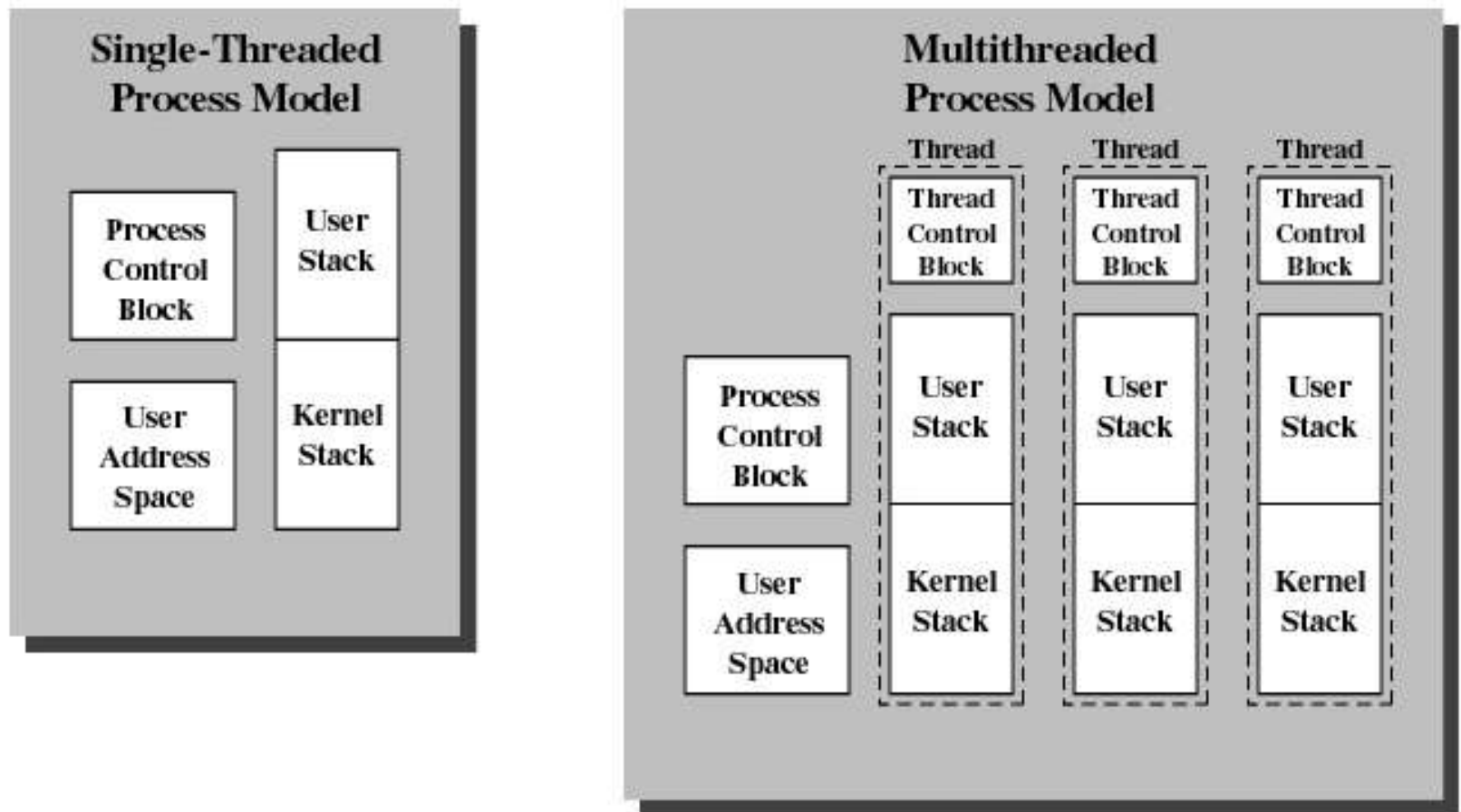


Figure 4.2 Single Threaded and Multithreaded Process Models

مزایای نخها:

- ایجاد یک نخ جدید در یک فرایند موجود از ایجاد یک وظیفه جدید بسیار کمتر وقت می برد.
- پایان دادن به یک نخ سریعتر است.
- تعویض دو نخ داخل یک فرایند کمتر وقت میگیرد.
- ارتباط بین برنامه های مختلف در حال اجرا. نظر به اینکه نخهای داخل یک فرایند در حافظه و پرونده ها شریک هستند، میتوانند بدون دخالت هسته با یکدیگر ارتباط برقرار کنند.

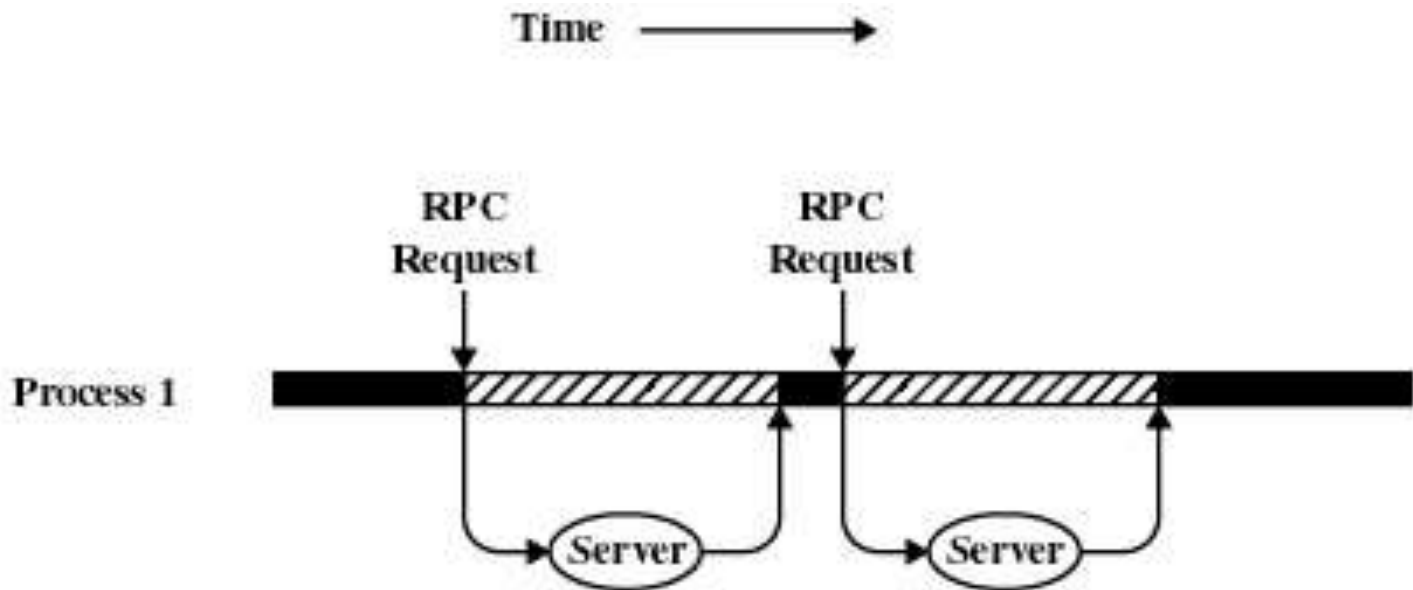
استفاده از نخها در یک سیستم تک کاربره چند پردازشی:

- چهار مثال مختلف از به کارگیری از نخها در یک سیستم تک کاربره چند پردازشی:
- کار پیش زمینه و پس زمینه
- پردازش ناهمگام
- موازی سازی و افزایش سرعت اجرا
- ساختار مؤلفه ای برنامه


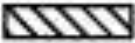

حالات و اعمال نخها:

- نخها نیز مانند فرایندها دارای حالات اصلی اجرا، آماده، مسدود هستند.
- تعلیق مستلزم مبادله فضای آدرسها، به خارج از حافظه اصلی است. از آنجا که تمام نخهای یک فرایند از فضای آدرسهای مشترک استفاده میکنند، باید در همان زمان تمام نخها به حالت معلق وارد شوند.
- پایان یافتن فرایند موجب پایان یافتن تمام نخهای آن فرایند میشود.
- چهار عمل نخها با تغییر حالت همراه است
 - **ایجاد (زایش):** با ایجاد یک فرآیند، نخ هم ایجاد می شود.
 - **مسدود شدن:** تا بروز حادثه نخ مسدود می شود.
 - **رفع مسدود بودن:** اتفاق حادثه لازم.
 - **پایان:** با تکمیل یک نخ، متن ثبات و پشته هایش آزاد می شود.

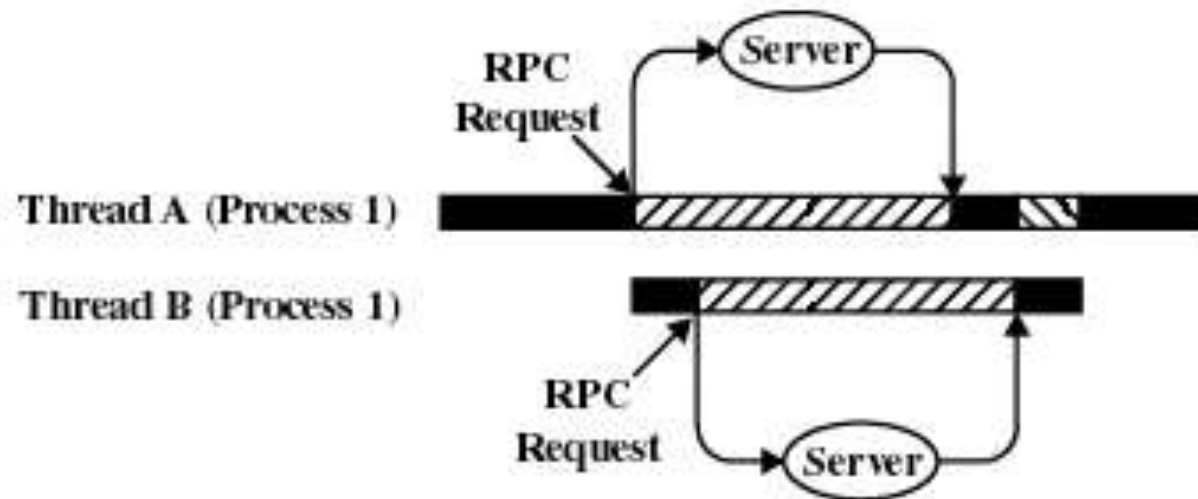
فراخوانی رویه از راه دور با استفاده از نخها:






(a) RPC Using Single Thread

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

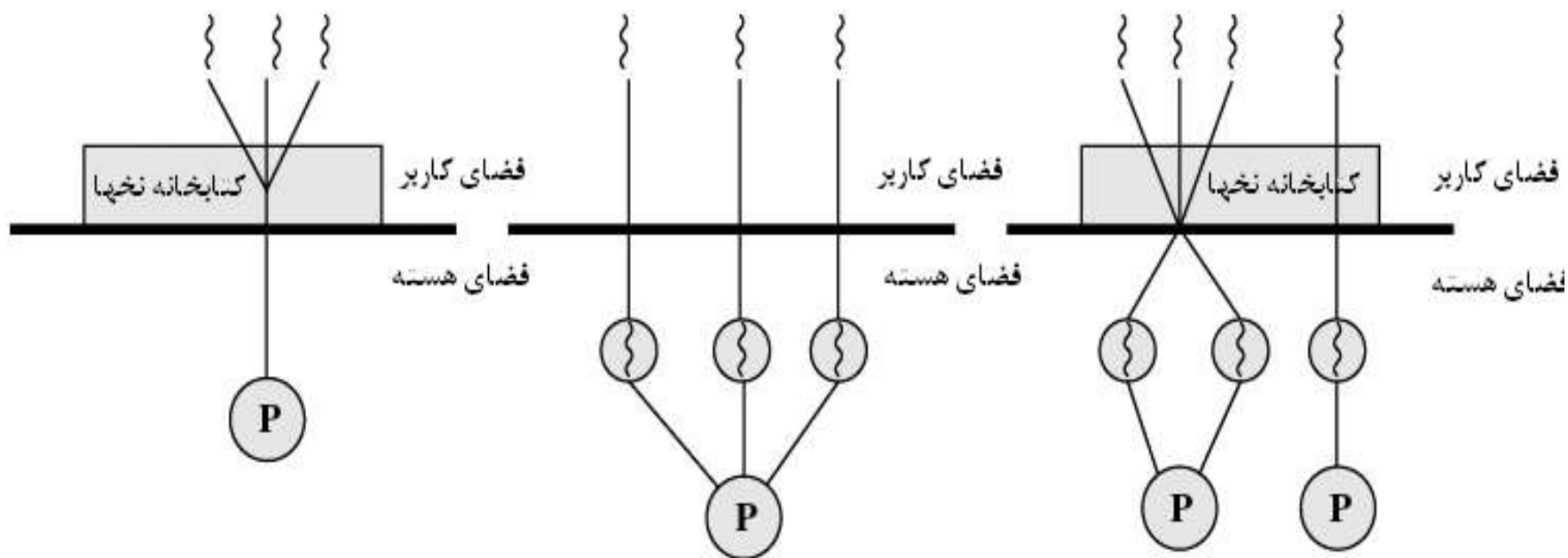
فراخوانی رویه از راه دور با استفاده از نخها:



(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

نمایی از نخهای سطح کاربر و هسته:



(a) سطح کاربر محض

(b) سطح هسته محض

(c) ترکیبی

{ نخ سطح کاربر { نخ سطح هسته P فرایند

نخهای سطح کاربر:

- تمام کار مدیریت نخها توسط کاربر انجام میگیرد.
- هسته از وجود نخها بی اطلاع است.
- با استفاده از کتابخانه نخها که مجموعه‌ای از روالها برای مدیریت نخهای سطح کاربر است، میتوان کاربرد ها را به صورت چندنخی برنامه سازی کرد.
- کتابخانه نخها شامل کد ایجاد و تخریب نخها، کد انتقال پیام ها و داده ها بین نخها، کد زمانبندی اجرای نخها، و کد ذخیره و بار کردن متن نخهاست.

نخهای سطح هسته:

- لینوکس، OS/2 و W2K نمونه هایی از این رویکرد اند.
- هسته متن اطلاعات را برای فرایندها و نخها نگهداری میکند.
- زمانبندی توسط هسته و بر مبنای نخ صورت میگیرد.

مزیت نخهای سطح کاربر:

- دو مزیت نخهای سطح کاربر نسبت به نخ های سطح هسته:
 - تعویض نخ به حالت ممتاز هسته نیاز ندارد، زیرا تمام ساختمان داده های مدیریت نخ در داخل فضای آدرس کاربر یک فرایند واحد قرار دارند. بنابراین برای مدیریت نخ نیازی به تعویض فرایند به حالت ممتاز هسته نیست. این، امکان صرفه جویی دو تعویض حالت (از کاربر به هسته و بالعکس) را میدهد.
 - کاربرد میتواند زمانبندی خاص داشته باشد. ممکن است کاربردی از یک الگوریتم زمانبندی نوبت گردشی ساده سود ببرد، در حالی که برای کاربرد دیگری یک الگوریتم زمانبندی مبتنی بر اولویت مفید تر باشد. میتوان الگوریتم زمانبندی را متناسب با کاربرد در نظر گرفت، بدون اینکه باری برای سیستم عامل داشته باشد.
 - نخ های سطح کاربر میتوانند بر روی هر سیستم عاملی اجرا شوند. هیچ تغییری در هسته برای حمایت از نخهای سطح کاربر لازم نیست. کتابخانه نخها، مجموعه ای از برنامه های سودمند در سطح کاربرد و مورد اشتراک تمام کاربرد هاست

عیب نخهای سطح کاربر:

- عیب نخهای سطح کاربر:

- در یک سیستم عامل متداول، اکثر فراخوانی های سیستم مسدود کننده هستند. بنابراین وقتی نخ یک فراخوانی سیستم را اجرا میکند نه تنها آن نخ بلکه تمام نخهای آن فرایند مسدود کننده اند.
- در یک راهبرد نخ سطح کاربر محض، کاربرد چند نخ نمیتواند از امتیازات چند پردازشی استفاده نماید. هسته در هر زمان یک فرایند را فقط به یک پردازنده نسبت میدهد. بنابراین در هر لحظه فقط یک نخ داخل فرایند میتواند اجرا شود.

جلد کردن:

- یک راه حل برای غلبه بر مساله نخهای مسدود کننده جلد کردن است.
- جلد کردن یک فراخوانی سیستم مسدود را به یک فراخوانی سیستمی نامسدود تبدیل میکند. برای مثال نخ به جای فراخوانی مستقیم یک روال ورودی خروجی سیستم یک روال سطح کاربرد ورودی خروجی از جلد را فرا میخواند تا وضعیت مشغولی دستگاه را چک کند.

مزیت نخهای سطح هسته:

- این رویکرد هر دو اشکال نخهای سطح کاربر را رفع میکند.
- اولاً هسته میتواند به طور همزمان نخهای چندگانه یک فرایند واحد را روی پردازنده های متعدد زمانبندی نماید.
- ثانياً اگر نخى از يك فرایند مسدود است، هسته میتواند نخ دیگری از همان فرایند را زمانبندی نماید.
- امتیاز دیگر: خود روال های هسته نیز می توانند چند نخى باشند.

عیب نخهای سطح هسته:

- عیب نخهای سطح هسته در مقابل نخهای سطح کاربر:
- انتقال کنترل از یک نخ به نخ دیگر در داخل یک فرایند نیز نیازمند تغییر به حالت هسته است.
- زمانبندی نخها در سطح هسته انجام می شود و نمیتواند هر فرایند زمان بندی خاص خود را داشته باشد.

راهبرد ترکیبی:

- بعضی از سیستم های عامل ترکیبی از نخهای سطح کاربر و نخهای سطح هسته را ارائه میدهند.
- Solaris نمونه ای از این نوع سیستم عامل است.
- تولید نخ به طور کامل در فضای کاربر در داخل یک کاربرد صورت میگیرد.
- قسمت عمده زمانبندی و همگام سازی نخها در فضای کاربر صورت میگیرد.
- ترکیبی از امکانات نخ سطح کاربر و نخ سطح هسته را فراهم می کند.
- به تعداد (کمتر یا مساوی) از نخ های سطح هسته نگاشته می شود.
- نخ ها به موازات هم روی پردازنده های متعدد اجرا می شود.
- یک فراخوان مسدود کننده کل سیستم را مسدود نمی کند.

روابط بین نخها و فرایندها:

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

معماری پردازنده های موازی:

- جریان دستورالعمل واحد داده های واحد:
Single Instruction Single Data (SISD) •
- جریان دستورالعمل واحد داده های چند گانه:
Single Instruction Multiple Data (SIMD) •
- جریان دستورالعمل چند گانه داده های واحد:
Multiple Instruction Single Data (MISD) •
- جریان دستورالعمل چند گانه داده های چند گانه :
Multiple Instruction Multiple Data (MIMD) •

جریان دستورالعمل واحد داده های واحد :

- یک پردازنده جریان واحدی از دستورالعمل را برای عمل روی داده هایی که در یک حافظه قرار دارند اجرا میکند.

جریان دستورالعمل واحد داده های چندگانه:

- دستورالعمل واحدی از پردازنده اجرای همزمان تعدادی از عناصر پردازشی را، بر اساس پیشروی یکسان کنترل میکند. هر یک از عناصر پردازشی، حافظه مربوط به خود را (برای داده هایش) دارد. بنابراین هر دستورالعمل توسط پردازنده های مختلف روی مجموعه های متفاوتی از داده ها اجرا میشود.
- پردازنده های برداری و آرایه ای از این نوع هستند.

جریان دستورالعمل چندگانه داده های واحد:

- دنباله ای از داده ها به مجموعه ای از پردازنده ها که هر یک دنباله متفاوتی از دستورالعمل ها را اجرا میکنند، فرستاده میشود.
- این ساختار هیچگاه پیاده سازی نشده است.

جریان دستورالعمل چندگانہ داده های چندگانہ:

- 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:

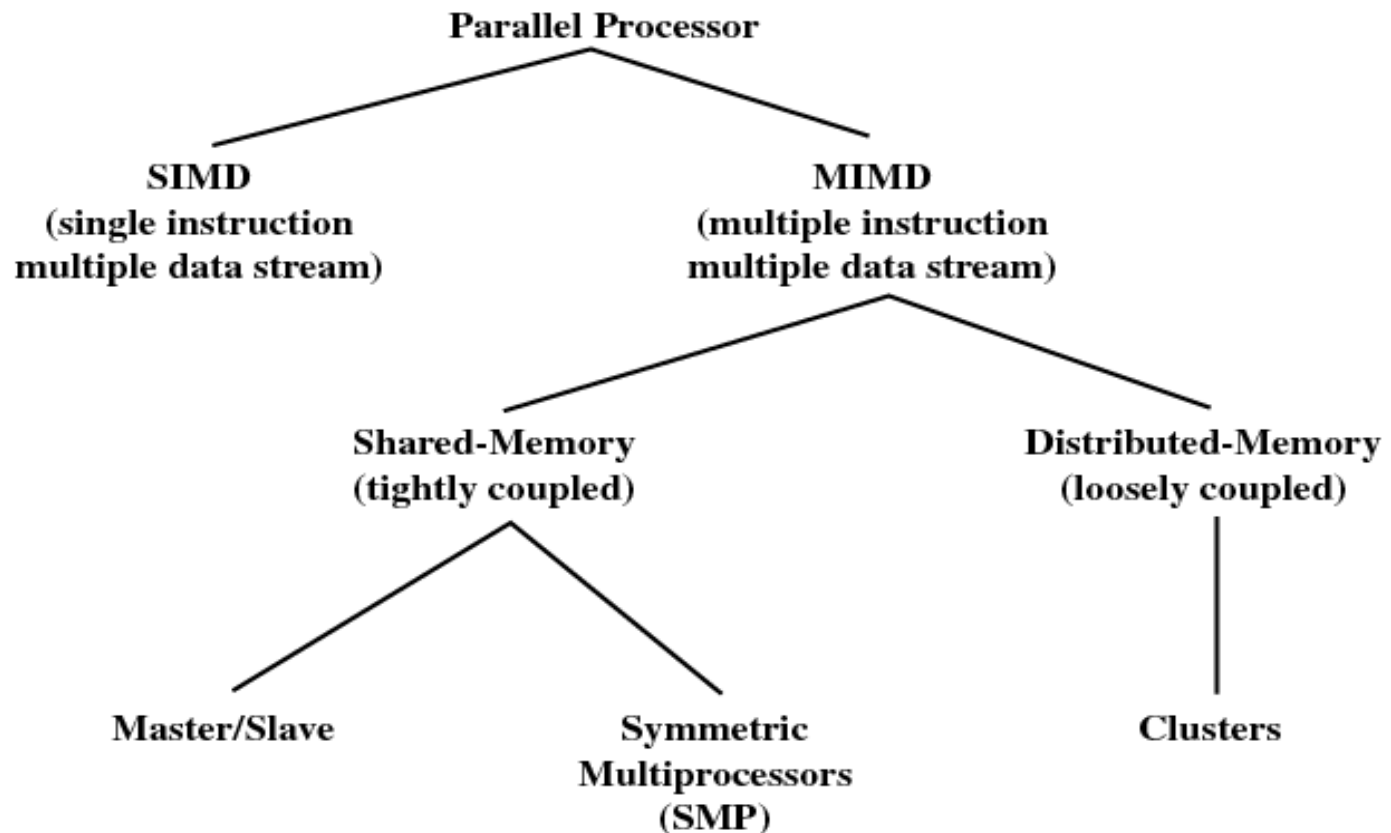


Figure 4.8 Parallel Processor Architectures

چند پردازشی متقارن:

- هر يك واحد كنترل، واحد حساب و منطق و ثباتهای خاص خود را دارند.
- به يك حافظه اصلی مشترك دسترسی دارد.
- پردازنده ها از طریق حافظه با يكديگر در ارتباط هستند.
- امکان تبادل مستقیم وجود دارد.
- هر پردازنده از طریق يك گذرگاه مشترك به يك حافظه اصلی متصل است.
- امکان دسترسی به بلوك ها مهيا است.
- تغییر در يك حافظه پنهان سبب تغییر در حافظه پنهان دیگر می شود. (وابستگی حافظه پنهان)
- با آن در سخت افزار بر خورد می شود.
- هسته میتواند بر روی هر پردازنده ای اجرا شود.
- نوعا هر پردازنده از نخها و فرایندهای موجود برای خود زمانبندی میکند.
- هسته میتواند به صورت فرایندهای چندگانه با نخهای چندگانه ساخته شود.

چند پردازشی متقارن:

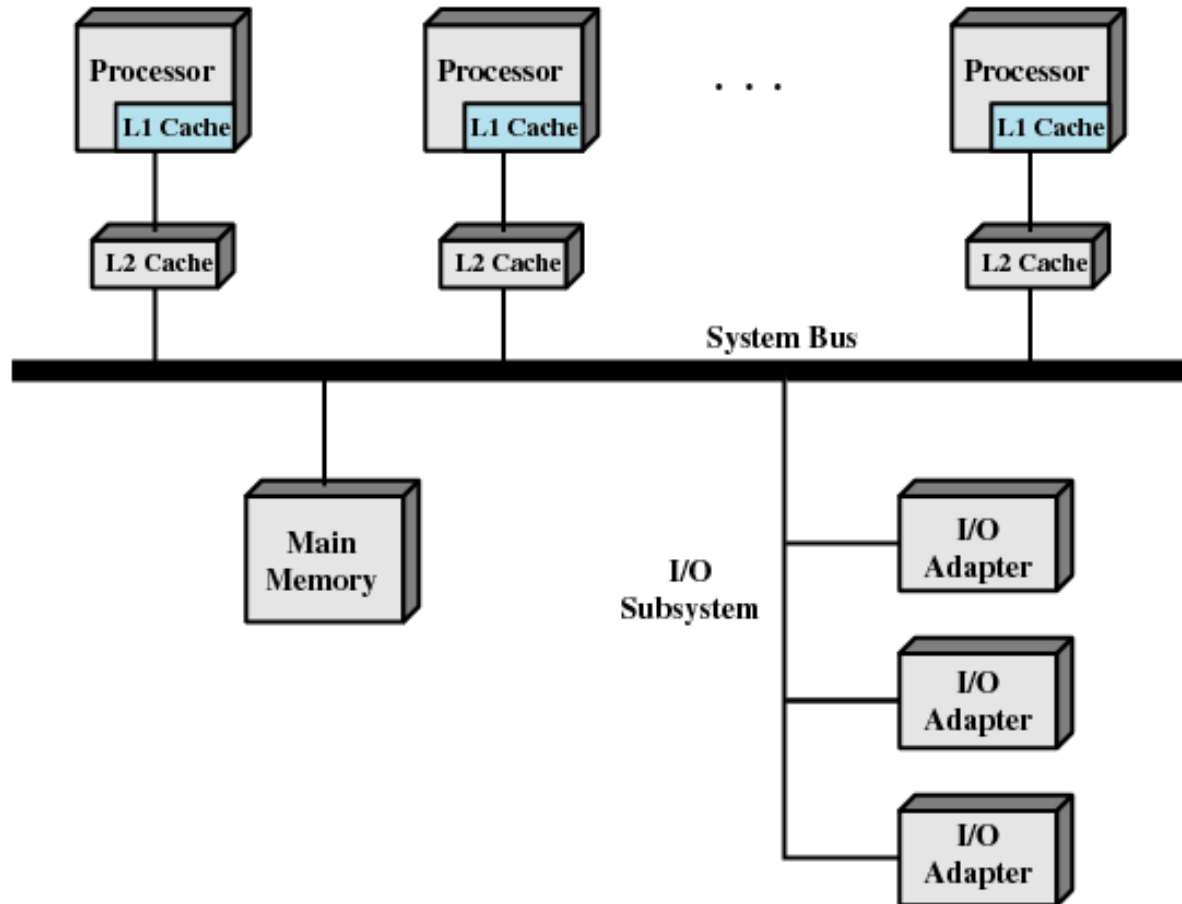


Figure 4.9 Symmetric Multiprocessor Organization

ملاحظات طراحی سیستم عامل چند پردازنده ای:

- هم زمانی فرآیند ها یا نخ ها: توسط هر یک از پردازنده ها انجام شود .

- زمانبندی: توسط هر یک از پردازنده ها انجام شود .

- همگام سازی: ابزاری است برای تنظیم انحصار متقابل و تنظیم حادثه ها.

ملاحظات طراحی سیستم عامل چند پردازنده ای:

- مدیریت حافظه: باید سخت افزار مناسب را بهره جوید.
- قابلیت اطمینان و تحمل خرابی: در برخورد با خرابی باید اقدام کند.

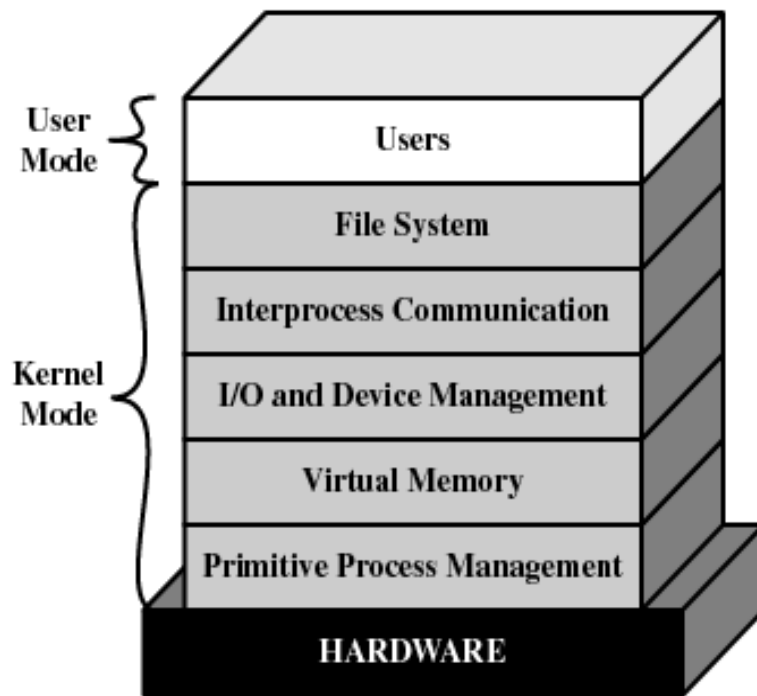
ریز هسته ها:

- ریز هسته سیستم عامل کوچکی است که بنیان لازم برای گسترشهای مولفه ای را به وجود می آورد.
- رویکرد ریز هسته با به کارگیری آن در سیستم عامل Mach مورد توجه عمومی قرار گرفت.
- سیستم عامل windows NT نیز دارای ریز هسته است که مدعی قابلیت حمل و مولفه ای بودن است.

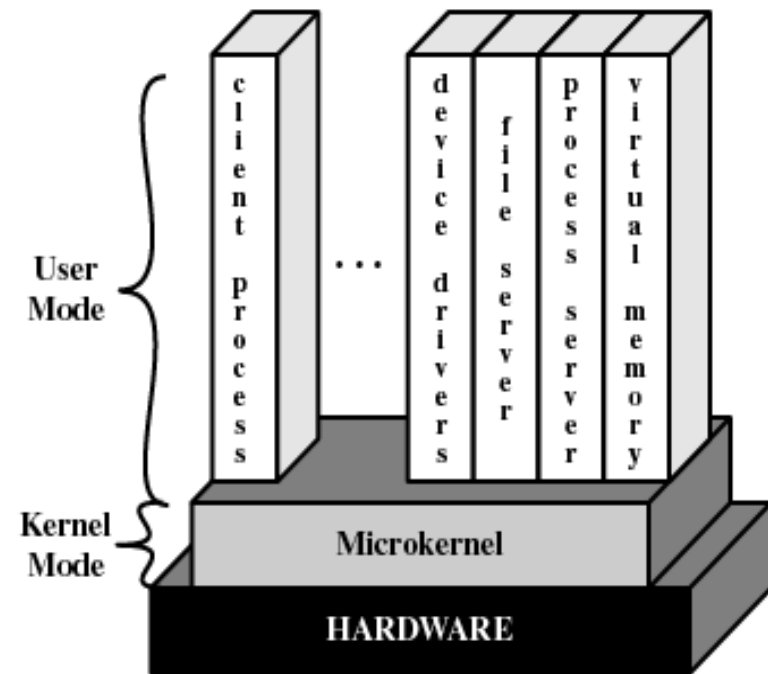
ریز هسته:

- فقط اعمال کاملاً اصلی سیستم عامل باید در هسته باشد
- خدمات و کاربردهایی که اصلی نیستند روی ریز هسته ساخته میشوند و در حالت کاربر اجرا میشوند.
- بخشهای زیر از سیستم عامل به عنوان زیر سیستم های خارجی اند.
 - گردانندگان دستگاه
 - سیستم های پرونده
 - مدیر حافظه مجازی
 - سیستم پنجره بندی
 - خدمات ایمنی

ریز هسته:



(a) Layered kernel



(b) Microkernel

Figure 4.10 Kernel Architecture

معماری ریز هسته:

- فلسفه ریز هسته این است که تنها اعمال کاملاً اصلی سیستم عامل باید در هسته باشد.
- کاربردها روی ریز هسته ساخته شده.
- بخشهایی از سیستم عامل که در خارج ریز هسته قرار دارند به صورت فرآیند خدمتگذار پیاده سازی می شوند.
- بر مبنای برابری و از طریق ریز هسته با هم ارتباط برقرار می کنند.
- ریز هسته به عنوان یک مبادله کننده پیام عمل می کند.

مزایای سازمان ریزهسته:

- **واسط های یکنواخت:** نیازی وجود ندارد که فرایندها بین خدمات سطح کاربر و سطح هسته تمایزی قائل شوند چرا که تمام این خدمات با استفاده از ارسال پیام فراهم میشود.
- **قابلیت گسترش:** اضافه کردن خدمات جدید یا حتی فراهم کردن خدمات چندگانه در یک محیط کاری را آسان میکند.

مزایای سازمان ریزهسته:

• **قابلیت انعطاف:** نه تنها ممکن است خصیصه های جدید به یک سیستم عامل اضافه شود بلکه ممکن است از خصوصیات موجود کاست تا به یک طراحی کوچکتر و کارآمد تر دست یافت.

• **قابلیت حمل:** همه یا حداقل اکثر کد مربوط به ویژگیهای خاص پردازنده در ریز هسته قرار دارد. بنابراین تغییرات لازم برای حمل سیستم عامل به پردازنده جدید کمتر میشود. و به قرار گیری در گروه های منطقی گرایش می یابند.

مزایای سازمان ریزهسته:

- **قابلیت اطمینان:** یک ریزهسته میتواند مورد آزمایش های سخت قرار گیرد. استفاده آن از تعداد کمی واسط های برنامه سازی کاربردی، مجال تولید کد با کیفیت برای خدمات سیستم عامل، در خارج از هسته را افزایش میدهد.

- **حمایت از سیستم توزیعی:** پیام گرایی ارتباطات ریزهسته، آن را به یک سیستم توزیعی گسترش میدهد.

- **حمایت از سیستم عامل شیءگرا:** میتوان از یک رویکرد شیء گرایی در نظام طراحی ریزهسته و ایجاد گسترش مولفه ای سیستم عامل استفاده کرد.

کارایی ریز هسته:

- به دلیل اینکه ریز هسته های مختلف طیفی از اعمال و اندازه را نشان می دهند، نمی توان هیچ قاعده محکم و سریعی در مورد اعمالی که ریز هسته باید ارائه کند و ساختار پیاده سازی آن اظهار داشت.
- ریز هسته خیلی کوچک سبب شده که کاهش کارایی را حذف و انعطاف و قابلیت اطمینان را بهتر کند.

طرح ریز هسته:

• ریز هسته مستقیماً به سخت افزار بستگی دارد و اعمال زیر را حمایت می کند:

۱- مدیریت حافظه اولیه.

۲- ارتباط بین فرآیندها.

۳- ورودی / خروجی.

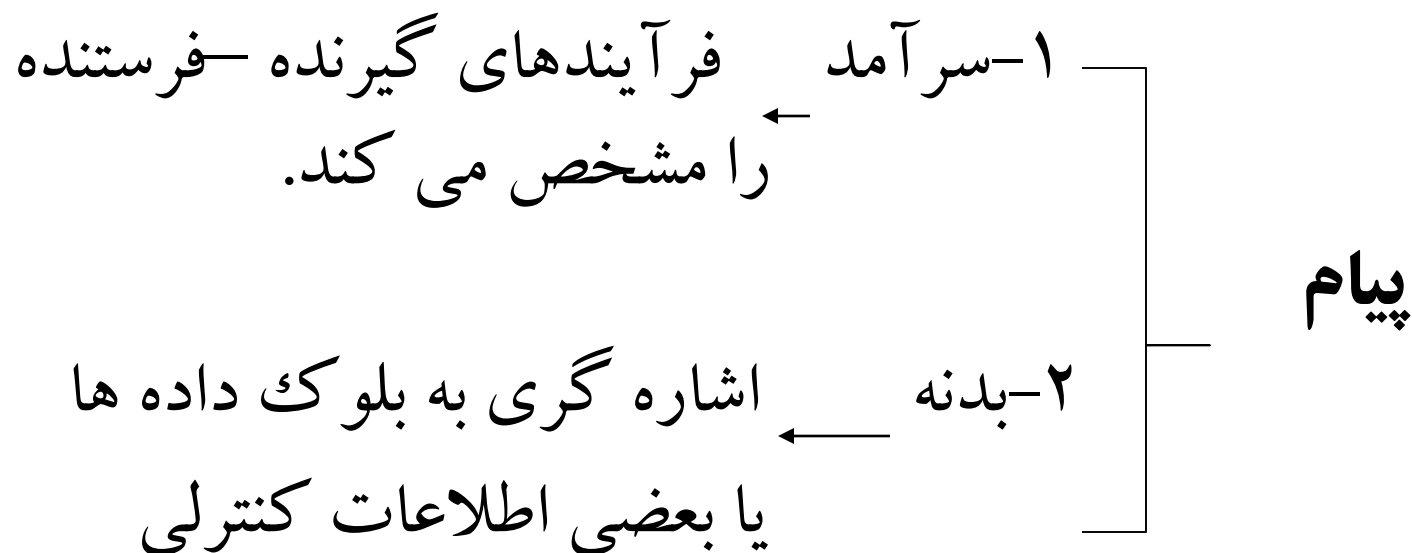
۴- مدیریت وقفه.

سه روش زیر برای حمایت مدیریت حافظه مجازی و صفحه بندی خارجی:

- **اهدای:** هر فرآیند می تواند تعدادی از صفحات خود را به دیگری دهد.
- **نگاشت:** یک فرآیند می تواند هر یک از صفحات خود را به فضای دیگری بنگارد.
- **مطالبه:** هر صفحه که قبلا استفاده شده را مطالبه کند.

ارتباط داخلی فرآیند:

• پیام شکل اصلی ارتباط بین فرایندها یا نخهاست.



مدیریت ورودی و خروجی و وقفه ها:

- اداره وقفه های سخت افزاری به صورت پیام و درگاه های ورودی و خروجی در فضاهاى آدرس میسر می شود.
- ریز هسته وقفه را تشخیص می دهد ولی آنها را اداره نمی کند.
- پیامی را برای فرآیند سطح کاربر مربوط به آن وقفه تولید می کند.