

# فصل سوم سیستم های توزیع شده: فرایندها

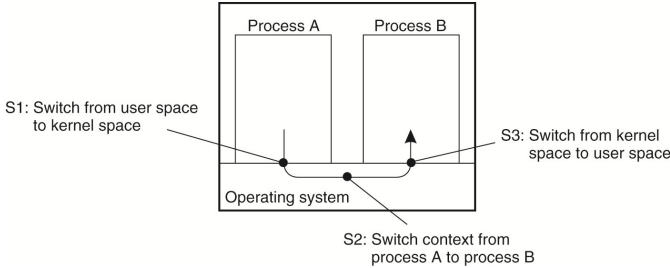
سیستمهای توزیع شده  
امیر مسعود رحمانی



## فرآیند چیست؟

یکی از مهمترین قسمتها در کامپیوتر فرآیند ها هستند، زیرا قسمت اعظم کارها توسط فرآیند ها صورت می گیرد.  
فرآیند یک برنامه در حال اجرا است! که معمولاً دارا مالکیت منابع خاصی است.  
سیستم عامل برای فرآیند یک PCB (که همه اطلاعات فرآیند در آن است) ایجاد می کند.  
برای امکان چندبرنامه ای سیستم عامل بین فرآیند ها سوییچ می کند.

## تعویض فرآیند



Context switching as the result of IPC.

## نخ چیست؟

- هر نخ شامل مجموعه ای از دستورات است.
  - نخها قسمتی از یک برنامه هستند و یک فرآیند در حال اجرا می تواند یک یا چند کار داشته باشد.
  - دلیل استفاده از نخ ها
- در حالتی که در یک فرآیند چند نخ همزمان اجرا شوند، امکان فراهم آوردن پردازش موازی و همچنین کار با برنامه های بزرگ ساده تر می شود.

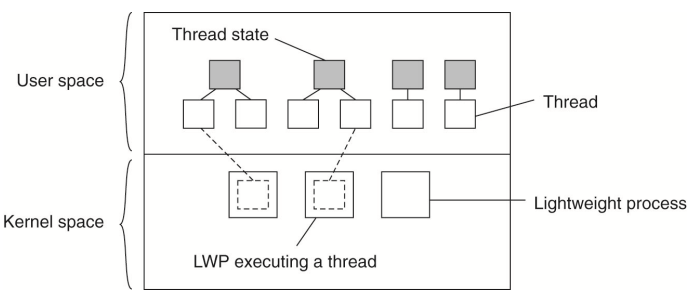
مزایا و معایب نخ سطح کاربر

- ایجاد و تخریب نخ در این سطح ارزان است.
- غالباً سوئیچ کردن روی اجرای نخ دیگر فقط می تواند در چند دستورالعمل انجام گیرد (چون از حافظه مشترک استفاده می کنند).
- با بکار گیری **System call** مسدود کننده، بلافاصله کل فرآیندی را که نخ به آن تعلق دارد مسدود خواهد کرد و در نتیجه همه نخ های دیگر این فرآیند مسدود خواهند شد.

پیاده سازی نخ

- عموماً نخ ها به شکل بسته کوچک نخی ارائه می شوند با قابلیت های  
ایجاد  
اتمام  
انحصار استفاده از داده و همزمانی  
مدیریت خطا
- پیاده سازی یک بسته کوچک نخی  
سطح کاربر  
سطح هسته  
ترکیبی

پیاده سازی نخ ترکیبی



ترکیب فرایندهای سبک وزن سطح هسته و نخ های سطح کاربر

مزایا و معایب نخ سطح هسته

مشکلات سطح کاربر را ندارد.  
هزینه برای هر عمل نخ (ایجاد، حذف، همزمانی و غیره) که به **System call** نیاز دارند بالا است.  
سوئیچ کردن برای اجرای نخ می تواند به گرانی سوئیچ کردن برای فرآیندهای دیگر باشد.



نخ ها در سیستم های توزیع شده

- در سیستم های توزیع شده معمولاً از فرآیندهای چند نخي در سمت سرویس گیرنده و سرویس دهنده استفاده می شود.
- سرویس گیرنده های چند نخي (Multithreaded Clients)
- سرویس دهنده های چند نخي (Multithreaded Servers)

مزایا و معایب نخ ترکیبی

- هنگام System call مسدود کننده اگر LWP کافی باشد کل فرآیند مسدود نمی شود.
- Application اطلاعی در مورد LWP ها نداشته و فقط نخهای سطح کاربر را می بیند.
- LWP ها به آسانی در محیط چند پردازشی مورد استفاده قرار می گیرند.
- ایجاد و حذف LWP ها به گرانی سطح هسته است.

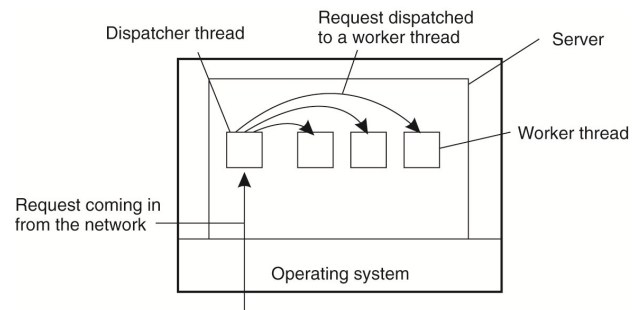
مثالی از سرویس دهنده چند نخي

- سرویس دهنده فایل چند نخي
- مدل توزیع کننده/کارگر: که در آن فرآیند سرویس دهنده یک نخ توزیع کننده دارد و چند نخ کارگر. هر درخواستی که به سرویس دهنده می رسد، توسط توزیع کننده دریافت می شود و بین کارگران تقسیم می شود.
- همزمانی ارسال فایل ها برای چند کاربر

مثالی از سرویس گیرنده چند نخي

- مرور گر وب چند نخي
- به محض اینکه فایل HTML اصلی واکنشی شد، در صورت وجود عکس در فایل HTML یک نخ ایجاد شده و هر نخ یک اتصال جداگانه به سرویس دهنده محل قرار گرفتن عکس ایجاد می کند و داده ها را دریافت می کند.
- کاربر با کارایی (سرعت) بالاتری صفحات وب را دریافت می کند.
- همزمانی دریافت عکس ها و متن

سرویس دهنده چند نخه توزیع کننده/کارگر



یک سرور چند نخه شده که در مدل کارگر / توزیع کننده سازماندهی شده است

سیستمهای توزیع شده

۱۳

روشهای ایجاد سرویس دهنده

Model	Characteristics
Threads	Parallelism, blocking system calls
Single-threaded process	No parallelism, blocking system calls
Finite-state machine	Parallelism, nonblocking system calls

بر اساس Event کار می کند.  
برنامه نویسی آن تا حدودی سخت است.

بیکاری پردازنده  
تعداد درخواستهای کمتری در هر ثانیه

سیستمهای توزیع شده

۱۴

سرویس گیرنده ها در  
سیستم توزیع شده

سیستمهای توزیع شده

۱۵

سرویس گیرنده ها در سیستم توزیع شده

۱- واسطه های کاربر شبکه شده  
(Networked User Interfaces)

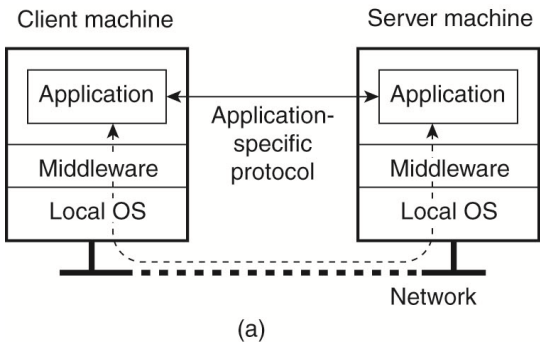
۲- نرم افزار طرف سرویس گیرنده برای شفافیت توزیع  
شده  
(Client-Side Software for Distribution Transparency)

سیستمهای توزیع شده

۱۶



### Networked User Interfaces (1)

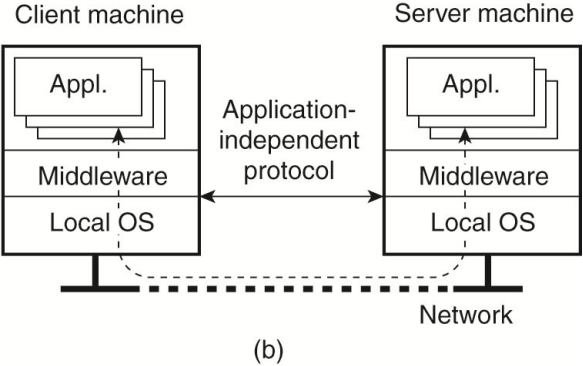


(a) A networked application with its own protocol

سیستمهای توزیع شده

۱۷

### Networked User Interfaces (2)



(b) A general solution to allow access to remote applications

سیستمهای توزیع شده

۱۸

### ۱- واسطه های کاربر شبکه شده

کار اصلی اکثر سرویس گیرنده ها، تعامل با کاربر انسانی و سرویس دهنده های راه دور است. پشتیبانی از واسطه کاربر، خصوصیات کلیدی بیشتر سرویس گیرنده ها است. سیستم XWindow مثالی از واسطه کاربر گرافیکی است.

سیستمهای توزیع شده

۱۹

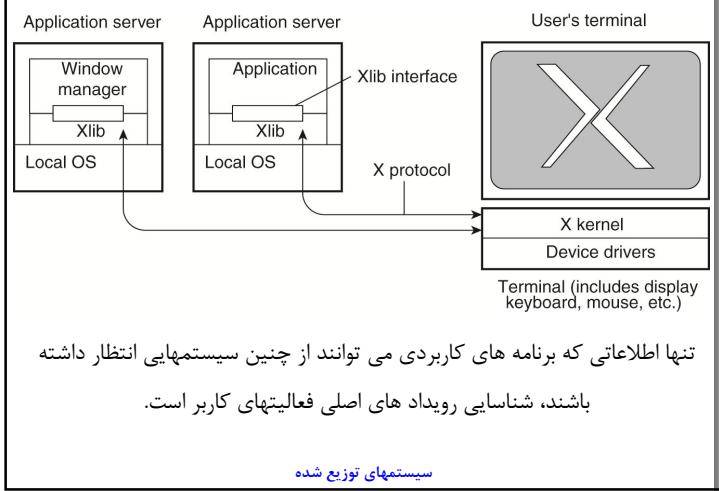
### سیستم XWindow

- برای کنترل ترمینال های طرح bit-map بکار می رود.
- X می تواند بعنوان آن بخش از سیستم عامل در نظر گرفته شود که ترمینال را کنترل می کند.
- بسیار به سخت افزار وابسته است.
- X دو نوع برنامه های کاربردی را از هم متمایز می کند:
  - ۱- برنامه های کاربردی نرمال
  - ۲- مدیران ویندوز

سیستمهای توزیع شده

۲۰

ساختار اصلی سیستم XWindow



۲- نرم افزار طرف سرویس گیرنده برای شفافیت توزیع شده (۱)

- نرم افزار سرویس گیرنده شامل چیزهای بیشتری از واسطه های کاربر است.
  - مثال
- ماشین های خود پرداز (ATM)  
ثبت کننده های چک  
بار کد خوان ها

سیستمهای توزیع شده

۲۲

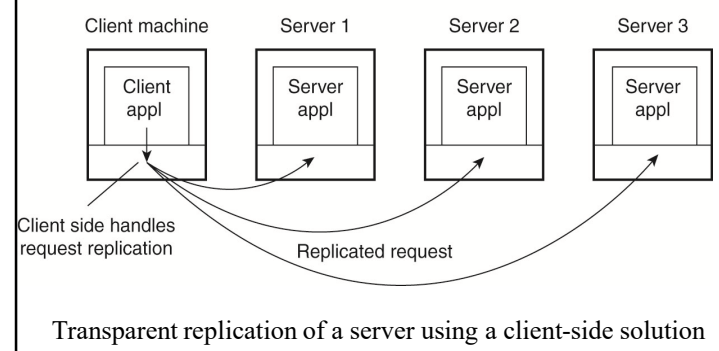
۲- نرم افزار طرف سرویس گیرنده برای شفافیت توزیع شده (۲)

- یک سرویس گیرنده نباید از اینکه در حال ارتباط با پردازش های راه دور است آگاه شود.
- در بدترین حالت، برنامه های کاربردی سرویس گیرنده ممکن است متوجه افت موقتی قابلیت اجرا بشود.
- خیلی از سیستمهای توزیع شده شفافیت تکثیر را بوسیله نرم افزار طرف سرویس گیرنده پیاده سازی می کنند.

سیستمهای توزیع شده

۲۳

۲- نرم افزار طرف سرویس گیرنده برای شفافیت توزیع شده (۳)



سیستمهای توزیع شده

۲۴



# سرویس دهنده ها در سیستم توزیع شده

سیستمهای توزیع شده

۲۵

## سرویس دهنده ها (۱)

یک سرویس دهنده فرایندی است که سرویس خاصی را به نفع یک مجموعه از سرویس گیرنده پیاده سازی می کند.  
۱- انواع سرویس دهنده ها از دیدگاه روش پیاده سازی:

### تکراری (Iterative Server)

خود سرویس دهنده درخواست را اداره نموده و در صورت لزوم پاسخی را به سرویس گیرنده می فرستد.

### همروند (Concurrent Server)

همانند چند نخي ها، درخواست های رسیده توسط سرویس دهنده اداره نمی شود بلکه آن را به فرایند یا نخ دیگری می فرستد و خود منتظر درخواست بعدی می شود.

سیستمهای توزیع شده

۲۶

## سرویس دهنده ها (۲)

۲- انواع سرویس دهنده ها از دیدگاه نگهداری وضعیت  
سرویس گیرنده  
Stateless Server

در Stateless هیچ وضعیتی از سرویس گیرنده نگهداری نمی شود و در صورت خرابی یا قطعی سرویس دهنده عملیات از اول شروع می شود.

### Stateful Server

در Stateful حالت و وضعیت سرویس گیرنده را نگهداری می کند.

### Soft state Server

در Soft State تا مدت محدودی اطلاعات سرویس گیرنده نگهداری می شود.

سیستمهای توزیع شده

۲۷

## سرویس دهنده ها (۳)

۳- انواع سرویس دهنده ها از دیدگاه نوع ارتباط  
End point port

در تمامی موارد سرویس گیرنده در خواست ها را به یک نقطه انتهایی به نام پورت در ماشینی که سرویس دهنده اجرا می شود می فرستد و هر سرویس دهنده به نقاط انتهایی گوش می کند.

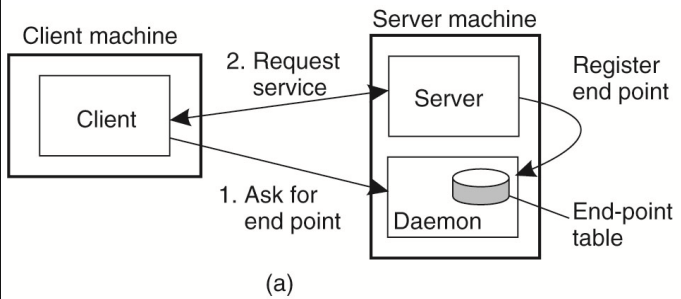
### Super server

دارای فقط یک سرویس دهنده است. در واقع ایجاد این ارتباط را بر عهده می گیرد و درخواست سرویس گیرنده را گرفته و به سرویس دهنده اختصاص میدهد.

سیستمهای توزیع شده

۲۸

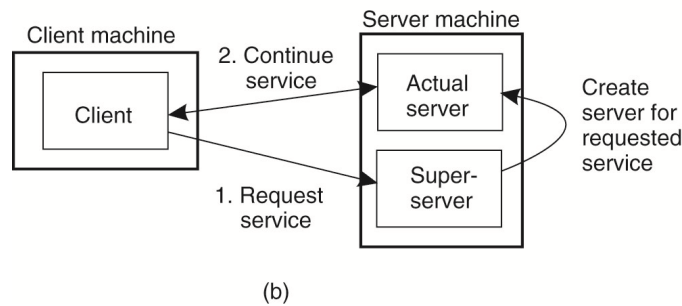
Client-to-server binding using a daemon



سیستمهای توزیع شده

29۳۹

Client-to-server binding using a Super server



سیستمهای توزیع شده

30۳۰

مهاجرت کد در سیستمهای توزیع شده

Code Migration

سیستمهای توزیع شده

۳۱

Reasons Migrating Code (1)

- So far, we have been passing data in distributed system
- Passing programs, some even while in execution, simplifies the design of distributed systems
- Dynamic downloading of client-server software, the software does not have to be preinstalled. i.e. Install on demand
- Harder in heterogeneous systems

سیستمهای توزیع شده

۳۲



### Migrating Code Examples (2)

- **Example 1: (Send Client code to Server)**  
Consider a client-Server system where server holds a huge database. If a client application needs to perform many database operations, it may be better to ship part of the client application to the server and server sends only the results across the network.
- **Example 2: (Send Server code to Client)**  
Data validation at the Database level: In many interactive DB applications, clients need to fill in forms that are subsequently translated into a series of DB operation where validation at server side is required.

سیستمهای توزیع شده

۳۳

### Migrating Code Examples (3)

- **Example 3:**  
System administrator may be forced to shut down a server but does not want to stop the running processes (e.g. to change a part or even permanent shut down)
- **Example 4:**  
Temporarily freeze an environment, move to another machine and unfreeze (e.g. for debugging sever production issues)
- **Example 4:**  
Parallel programming (e.g. Chess)

سیستمهای توزیع شده

۳۴

### مزایای مهاجرت کد

- ۱- اجرا شدن به صورت موازی
- ۲- استفاده از قدرت پردازنده ها و بالا رفتن کارایی
- ۳- توازن و تعادل بار
- ۴- انعطاف پذیری
- ۵- کاهش ارتباطات شبکه

سیستمهای توزیع شده

۳۵

### معایب مهاجرت کد

- ۱- امنیت پایین می آید (زیرا این کد ممکن است مخرب باشد)
- ۲- خیلی وقتها مهاجرت کد یا پروسس به جای بهبود کارایی باعث کاهش کارایی می شود ( مثلا بدلیل وجود هزینه های ارتباطی)

سیستمهای توزیع شده

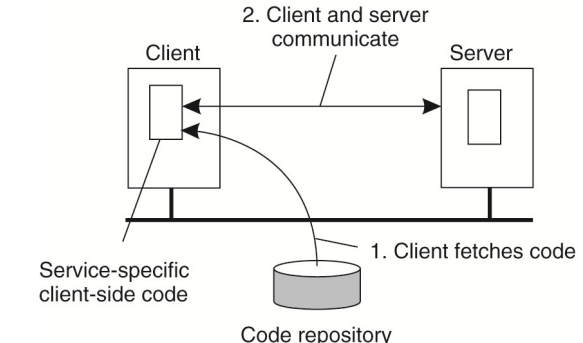
۳۶

مهاجرت کد

انتقال کد های برنامه به دلایلی در سیستم های توزیع شده صورت می پذیرد.  
انتقال کدهای برنامه جایگزین انتقال فرایندها شده است (Client-Side Programming)  
در انتقال فرایندها کل یک فرایند از یک ماشین به ماشین دیگر منتقل می شود.

سیستمهای توزیع شده

مهاجرت کد



اصول پیکر بندی خودکار یک کلاینت برای برقراری ارتباط با یک سرور. کلاینت ابتدا نرم افزار لازم را واگشی می کند و سپس آن سرور را فراخوانی می کند.

سیستمهای توزیع شده

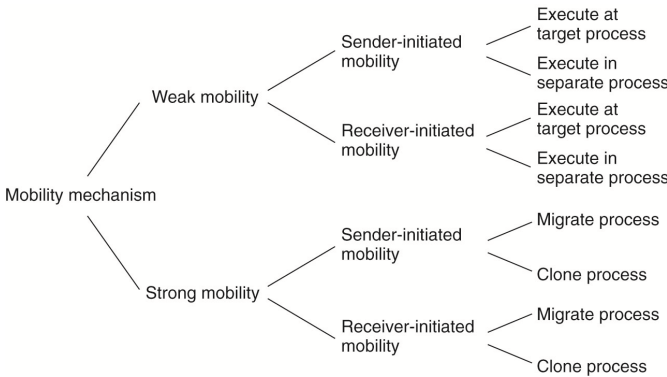
انواع مهاجرت فرایند

اجزای یک فرایند

- ۱- قطعه کد: مجموعه دستورالعمل ها که فرایند در حال اجرا را می سازد.
- ۲- قطعه منبع: شامل ارجاع به منابع خارجی مورد نیاز فرآیند.
- ۳- قطعه اجرا: جایی برای ذخیره حالت فعلی اجرای فرایند (PCB)

سیستمهای توزیع شده

مدلهای مهاجرت پروسس



سیستمهای توزیع شده



**جابهجایی ضعیف Weak mobility**

در این حالت تنها قطعه کد قابل جابهجایی است و قطعه اجرا منتقل نمی شود.  
ویژگی این است که همیشه فرایند از وضعیت ابتدایی شروع به کار می کند.  
مثال: اپلت های جاوا  
مهمترین حسن این روش سادگی آن است.

سیستمهای توزیع شده

۴۱

**جابهجایی قوی Strong mobility**

در این حالت علاوه بر قطعه کد، قطعه اجرایی نیز قابل انتقال است.  
امکان اجرای ادامه فرایند مهاجرت کننده را دارد (می تواند از ادامه اش اجرا شود).  
این مدل قوی تر و در عین حال پیچیده تر است.  
این مدل زیاد کاربردی نیست.

سیستمهای توزیع شده

۴۲

**آغاز شده توسط فرستنده Sender-initiated**

جابهجایی توسط ماشینی که کد روی آن است آغاز می شود.  
مثال  
- ارسال برنامه جستجو در اینترنت  
- برنامه با قابلیت پردازش موازی (شطرنج)

سیستمهای توزیع شده

۴۳

**آغاز شده توسط گیرنده Receiver-initiated**

در این مدل این ماشین مقصد است که کار جابهجایی را آغاز می کند.  
این مدل معمولاً ساده تر از مدل قبلی پیاده سازی می شود.  
مثال  
- اپلت های جاوا

سیستمهای توزیع شده

۴۴



مهاجرت قطعه منابع: اتصال فرایند به منبع

- ۱- By Value: فرایند با مقدار منابع سروکار دارد (فرایندی که از پایگاه داده یا فایل می خواند).
- ۲- By Type: فرایند از نوع خاصی از منبع استفاده می کند. مثلاً مانیتور
- ۳- By Identifier: فرایند که با شناسه سروکار دارد. مثلاً Port یا web service یا URL

مهاجرت قطعه منابع: اتصال منبع به ماشین

- ۱- Unattached (غیر متصل): منبع براحتی قابل جدا شدن از ماشین و انتقال است. مثل DLL و فایل کوچک
- ۲- Fastened (بسته شده): منبع قابل جدا شدن از ماشین و انتقال است ولی هزینه بر است. مثل پایگاه داده بزرگ
- ۳- Fixed (ثابت): منبع از ماشین جدا نمی شود. مثل پرینتر  
Local disk drives, communication ports

راه حل مهاجرت قطعه منابع (۱)

- GR منبع عمومی به گسترده گی سیستم ایجاد می کند.
- MV منبع را نقل مکان می دهد.
- CP مقدار منبع را کپی می کند.
- RB فرایند با منابع در دسترس محلی را دوباره اختیار می کند.

راه حل مهاجرت قطعه منابع

Resource-to-machine binding				
Process-to-resource binding		Unattached	Fastened	Fixed
	By identifier	MV (or GR)	GR (or MV)	GR
	By value	CP (or MV,GR)	GR (or CP)	GR
	By type	RB (or MV,CP)	RB (or GR,CP)	RB (or GR)
	GR	Establish a global systemwide reference		
	MV	Move the resource		
	CP	Copy the value of the resource		
	RB	Rebind process to locally-available resource		



## مهاجرت در سیستم های ناهمگن

در سیستمهای ناهمگن ممکن است انتقال به ماشینی صورت گیرد که ساختار متفاوتی (نرم افزاری و سخت افزاری) از ماشین مبدأ داشته باشد.

باید اطمینان وجود داشته باشد که پس از انتقال کد به ماشین مقصد، کد ها قابل اجرا باشند.

ممکن است نیاز به ترجمه مجدد کد ها وجود داشته باشد.

راه حل مهاجرت در سیستم های ناهمگن استفاده از **Middleware** ماشین مجازی مثل (JVM) است. که بصورت مجازی همه را همگن می کند.