
درس و کنکور سیستم عامل پیشرفته

تألیف:

دکتر رمضان عباس نژادورزی
دکتر مرتضی بابازاده
دکتر فاطمه عبدی سقاواز



فن آوری نوین

سرشناسه	عباس نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	درس و کنکور سیستم عامل پیشرفته/تالیف رمضان عباس نژاد ورزی، مرتضی بابازاده، فاطمه عبدی سقاوژ.
مشخصات نشر	بابل: فناوری نوین، ۱۳۹۶.
مشخصات ظاهری	۳۳۲ص.
شابک	۹۷۸-۶۰۰-۷۲۷۲-۰۲-۲
وضعیت فهرست نویسی	فیا
موضوع	سیستم های عامل توزیع شده (کامپیوتر) -- راهنمای آموزشی (عالی)
موضوع	(Distributed operating systems (Computers)-- Study and teaching (Higher
موضوع	سیستم های عامل توزیع شده (کامپیوتر) -- مسائل، تمرین ها و غیره (عالی)
موضوع	(Distributed operating systems (Computers) -- Problems, exercises, etc. (Higher
موضوع	سیستم های عامل توزیع شده (کامپیوتر) -- آزمون ها و تمرین ها (عالی)
موضوع	Distributed operating systems (Computers) Examinations, questions, etc. (Higher
شناسه افزوده	بابازاده، مرتضی، ۱۳۶۰ -
شناسه افزوده	عبدی سقاوژ، فاطمه، ۱۳۵۶ -
رده بندی کنگره	۱۳۹۶ ۹/۷۶QA ع۲۳۶پ/
رده بندی دیویی	۳۶/۰۰۴
شماره کتابشناسی ملی	۴۸۷۲۳۷۸



www.fanavarienovin.net

بابل، کدپستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

درس و کنکور سیستم عامل پیشرفته

تألیف: رمضان عباس نژاد ورزی - مرتضی بابا زاده - فاطمه عبدی سقاوژ

نوبت چاپ: چاپ اول

سال چاپ: پاییز ۱۳۹۶

شمارگان:

قیمت: ۳۲۰۰۰ تومان

نام چاپخانه و صحافی:

شابک: ۹۷۸-۶۰۰-۷۲۷۲-۰۲-۲

نشانی ناشر: بابل، چهارراه نواب، کاظم بیگی، جنب مسجد منصور کاظم بیگی، طبقه همکف

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

تهران، خ اردیبهشت، نش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

فصل اول: آشنایی با سیستم‌های توزیع شده

۱-۱. تعریف سیستم توزیع شده.....	۱۱
۱-۲. اهداف سیستم‌های توزیع شده.....	۱۱
۱-۲-۱. برقراری ارتباط آسان بین منابع و کاربران.....	۱۱
۱-۲-۲. شفافیت.....	۱۲
۱-۲-۳. باز بودن.....	۱۴
۱-۲-۴. توسعه پذیری.....	۱۵
۱-۳. تکنیک‌های توسعه پذیری.....	۱۶
۱-۳-۱. مخفی کردن تأخیر ارتباطات.....	۱۷
۱-۳-۲. توزیع.....	۱۸
۱-۳-۳. تکثیر.....	۱۸
۱-۴. سیستم‌های چند پردازنده‌ای.....	۱۹
۱-۴-۱. سیستم‌های چند پردازنده‌ای UMA با سوئیچ.....	۱۹
۱-۵. سیستم‌های با دسترسی یک نواخت به حافظه اشتراکی.....	۲۱
۱-۶. سیستم‌های چند کامپیوتری.....	۲۱
۱-۷. سیستم‌های چند کامپیوتری مبتنی بر سوئیچ.....	۲۲
۱-۸. حافظه توزیع شده‌ی اشتراکی.....	۲۳
۱-۹. مقایسه انواع سیستم‌ها.....	۲۵
۱-۹-۱. انواع سیستم عامل‌های توزیع شده.....	۲۵
۱-۹-۲. مقایسه سیستم‌ها با یکدیگر.....	۲۹
۱-۹-۳. تله‌ها (دام‌ها) در سیستم‌های توزیع شده.....	۲۹
۱-۱۰. انواع سیستم‌های توزیع شده.....	۲۹
۱-۱۰-۱. سیستم‌های محاسباتی توزیع شده.....	۳۰
۱-۱۰-۲. سیستم‌های اطلاعاتی توزیع شده.....	۳۲
۱-۱۰-۳. سیستم‌های فراگیر توزیع شده.....	۳۴
۱-۱۱. شبکه‌های حسگر.....	۳۶
۱-۱۲. سوالات چهار گزینه‌ای.....	۳۷
۱-۱۳. پاسخ تشریحی سوالات چهار گزینه‌ای.....	۴۵

فصل دوم: معماری‌های نرم افزار

- ۲-۱. سبک معماری..... ۵۱
- ۲-۱-۱. معماری لایه‌ای..... ۵۱
- ۲-۱-۲. معماری مبتنی بر شیء..... ۵۱
- ۲-۱-۳. معماری مبتنی بر داده مرکزی (داده محور)..... (۵۲)
- ۲-۱-۴. معماری مبتنی بر رویداد..... ۵۳
- ۲-۲. انواع معماری سیستم‌های توزیع شده..... ۵۳
- ۲-۲-۱. معماری متمرکز..... ۵۴
- ۲-۲-۲. معماری نامتمرکز..... ۶۷
- ۲-۲-۳. معماری ترکیبی (مختلط)..... ۶۴
- ۲-۳. خودگردانی..... ۶۷
- ۲-۳-۱. مدل کنترل بازخورد..... ۶۷
- ۲-۳-۲. روش‌های کلی نرم افزار تطبیقی..... ۶۸
- ۲-۴. سوالات چهار گزینه‌ای..... ۷۱
- ۲-۵. پاسخ تشریحی سوالات چهار گزینه‌ای..... ۷۹

فصل سوم: فرآیند

- ۳-۱. نخ‌ها..... ۸۴
- ۳-۱-۱. فرآیندهای میکرو وزن..... ۸۷
- ۳-۱-۲. سابقه فعالیت زمان بندی..... ۸۸
- ۳-۲. فراخوانی سیستمی..... ۸۹
- ۳-۳. کاربرد نخ در سیستم‌های توزیع شده..... ۸۹
- ۳-۴. مجازی سازی..... ۹۰
- ۳-۵. حالت‌ها در سرویس دهنده..... ۹۳
- ۳-۶. دلایل مهاجرت کد..... ۹۴
- ۳-۶-۱. مدل‌های مهاجرت کد..... ۹۴
- ۳-۶-۲. مهاجرت و منابع محلی..... ۹۶
- ۳-۷. سوالات چهار گزینه‌ای..... ۹۷
- ۳-۸. پاسخ تشریحی سوالات چهار گزینه‌ای..... ۱۵۲

فصل چهارم: ارتباطات

۱-۴. پروتکل های لایه ای.....	۱۰۶
۱-۴-۱. تفاوت پروتکل اتصال گرا و بدون اتصال عبارت اند از:.....	۱۰۷
۲-۴. لایه ها.....	۱۰۷
۱-۴-۲. لایه های پایین.....	۱۰۷
۳-۴. پروتکل های سطح بالا.....	۱۰۹
۴-۴. پروتکل های میان افزار.....	۱۱۰
۵-۴. انواع ارتباطات.....	۱۱۰
۶-۴. فراخوانی رویه های راه دور.....	۱۱۱
۷-۴. اجزای RPC.....	۱۱۲
۸-۴. انواع ارسال پارامترها.....	۱۱۴
۹-۴. اشیا توزیع شده.....	۱۱۷
۱۰-۴. مفهوم RMI.....	۱۱۹
۱۱-۴. ارتباط پیام گرا (مبتنی بر پیام).....	۱۲۰
۱۲-۴. ارتباط ناپایدار پیام گرا.....	۱۲۱
۱-۴-۱۲. سوکت بر کلی.....	۱۲۲
۲-۴-۱۲. واسطه مبادله پیام.....	۱۲۲
۳-۴-۱۲. ارتباط پایدار پیام گرا.....	۱۲۳
۱-۴-۱۳. مدل صف بندی پیام.....	۱۲۳
۲-۴-۱۳. معماری کلی سیستم صف بندی پیام.....	۱۲۶
۳-۴-۱۳. کارگزاران پیام.....	۱۲۷
۴-۴-۱۴. ارتباط جریان گرا.....	۱۲۹
۵-۴-۱۵. جریان داده.....	۱۲۹
۶-۴-۱۶. یک معماری کلی.....	۱۳۱
۷-۴-۱۷. جریان ها و کیفیت سرویس.....	۱۳۱
۸-۴-۱۸. تأکیدات سرویس QOS.....	۱۳۱
۹-۴-۱۹. همگام سازی جریان.....	۱۳۳
۲۰-۴. ارتباط چندپخشی.....	۱۳۴
۲۱-۴. چندپخشی سطح کاربرد.....	۱۳۴
۲۲-۴. انتشار داده شایعه گرا.....	۱۳۴

- ۴-۲۳. شایعه پراکنی..... ۱۳۵
- ۴-۲۴. سؤالات چهار گزینه‌ای..... ۱۳۶
- ۴-۲۵. پاسخ تشریحی سؤالات چهار گزینه‌ای..... ۱۴۵

فصل پنجم: نام گذاری

- ۵-۱. نام‌ها، شناسه‌ها و آدرس‌ها..... ۱۵۱
- ۵-۲. انواع نام گذاری..... ۱۵۲
- ۵-۳. روش‌های مبتنی بر مکان خانگی..... ۱۵۵
- ۵-۴. جدول‌های درهم سازی توزیع شده (DHT)..... ۱۵۶
- ۵-۵. روش‌های سلسله مراتبی..... ۱۵۸
- ۵-۶. نام گذاری ساخت یافته..... ۱۶۰
- ۵-۷. فضای نام..... ۱۶۰
- ۵-۷-۱. تبدیل نام..... ۱۶۲
- ۵-۷-۲. پیاده‌سازی فضای نام..... ۱۶۵
- ۵-۷-۳. پیاده‌سازی تبدیل نام..... ۱۶۶
- ۵-۸. سیستم نام دامنه..... ۱۶۸
- ۵-۹. نام گذاری بر اساس خصیصه..... ۱۶۹
- ۵-۱۰. سؤالات چهار گزینه‌ای..... ۱۷۰
- ۵-۱۱. پاسخ تشریحی سؤالات چهار گزینه‌ای..... ۱۷۵

فصل ششم: همگام سازی

- ۶-۱. ساعت فیزیکی..... ۱۸۰
- ۶-۲. ساعت اتمی..... ۱۸۰
- ۶-۳. سیستم تعیین موقعیت جهانی..... ۱۸۱
- ۶-۴. الگوریتم همگام سازی در شبکه بی سیم (RBS)..... ۱۸۷
- ۶-۵. ساعت منطقی..... ۱۸۷
- ۶-۶. انحصار متقابل..... ۱۹۲
- ۶-۷. الگوریتم متمرکز..... ۱۹۳
- ۶-۸. الگوریتم توزیع شده..... ۱۹۴
- ۶-۹. الگوریتم نشانه حلقه..... ۱۹۶
- ۶-۱۰. مقایسه سه الگوریتم..... ۱۹۷

۱۹۷	۶-۱۱. الگوریتم‌های انتخابات
۲۰۰	۶-۱۲. الگوریتم انتخاب در محیط‌های بی‌سیم
۲۰۱	۶-۱۳. انواع تراکنش‌ها
۲۰۱	۶-۱۳-۱. تراکنش‌های تخت
۲۰۱	۶-۱۳-۲. تراکنش‌های تودرتو
۲۰۱	۶-۱۳-۳. تراکنش‌های توزیع شده
۲۰۲	۶-۱۴. پیاده‌سازی تراکنش‌ها
۲۰۵	۶-۱۵. سوالات چهارگزینه‌ای
۲۱۵	۶-۱۶. پاسخ تشریحی سوالات چهارگزینه‌ای

فصل هفتم: سازگاری و تکثیر

۲۲۳	۷-۱. دلایل تکثیر
۲۲۳	۷-۲. تکثیر به عنوان تکنیک توسعه پذیری
۲۲۴	۷-۳. سازگاری مبتنی بر داده
۲۲۵	۷-۴. سازگاری پیوسته
۲۲۶	۷-۵. فرضیه کانیت
۲۲۸	۷-۶. ترتیب سازگاری عملیات
۲۲۸	۷-۶-۱. سازگاری متوالی
۲۳۱	۷-۶-۲. سازگاری علی
۲۳۲	۷-۷. گروه‌بندی عملیاتی
۲۳۲	۷-۸. سازگاری در مقابل انسجام
۲۳۲	۷-۹. مدل‌های سازگاری مبتنی بر سرویس گیرنده
۲۳۳	۷-۹-۱. سازگاری نهایی

۲۳۴	۷-۹-۲. خواندن های یک نواخت.....
۲۳۵	۷-۹-۳. نوشتن های یک نواخت.....
۲۳۶	۷-۹-۴. خواندن نوشتن های تان.....
۲۳۶	۷-۹-۵. نوشتن پس از خواندن ها.....
۲۳۷	۷-۱۰. مدیریت تکثیر.....
۲۳۶	۷-۱۰-۱. مکان یابی سرویس دهنده تکثیر شده.....
۲۳۸	۷-۱۰-۲. مکان یابی و تکثیر محتوی.....
۲۳۹	۷-۱۱. نسخه های تکثیر شده سرویس دهنده آغاز گر.....
۲۴۰	۷-۱۲. مدل های ساز گاری داده محور.....
۲۴۱	۷-۱۳. پروتکل های مبتنی بر نسخه اصلی.....
۲۴۲	۷-۱۳-۱. پروتکل های نوشتن راه دور.....
۲۴۳	۷-۱۳-۲. پروتکل های نوشتن محلی.....
۲۴۳	۷-۱۴. پروتکل نوشتن تکراری.....
۲۴۳	۷-۱۴-۱. تکثیر فعال.....
۲۴۴	۷-۱۴-۲. پروتکل مبتنی بر حد نصاب.....
۲۴۵	۷-۱۵. سوالات چهار گزینه ای.....
۲۵۰	۷-۱۶. پاسخ تشریحی سوالات چهار گزینه ای.....
۲۵۵	پیوست الف: تست های تکمیلی.....
۲۹۵	پیوست ب: سوالات دکتری.....
۳۲۰	پیوست ج: پرسش های حل شده در سایت.....
۳۲۹	پیوست د: مباحث تکمیلی.....
۳۳۲	منابع:

مقدمه

امروزه با رشد رایانه و اینترنت، استفاده از سیستم‌های توزیع شده به خصوص رایانش ابری روز به روز در حال افزایش می‌باشد. به همین دلیل، سیستم‌های توزیع شده به عنوان یکی از دروس اصلی در دوره کارشناسی ارشد در وزارت علوم و دانشگاه آزاد اسلامی تدریس می‌شود. به طوری که در آزمون دکتری نرم‌افزار نیز از آن سوال می‌آید. به همین دلیل، کتاب حاضر تهیه گردید و در اختیار علاقه‌مندان قرار گرفته است.

این کتاب شامل مباحث زیر می‌باشد:

۱. مفاهیم سیستم‌های توزیع شده، معماری‌ها، فرآیندها.
 ۲. ارتباطات، نام‌گذاری، همگام‌سازی، تکثیر و سازگاری
 ۳. تحمل خطا، امنیت و مدیریت فایل در سیستم‌ها توزیع شده. البته تحمل خطا، امنیت و مدیریت فایل‌ها در سیستم‌های توزیع شده را می‌توانید از سایت انتشارات فناوری نوین دانلود کنید.
 ۴. حدود ۷۰۰ سوال چهار گزینه‌ای فراگیر و ترمی پیام نور با پاسخ تشریحی آن‌ها.
 ۵. سوالات دکتری سراسری سال‌های ۹۱، ۹۲، ۹۳، ۹۴ و سوالات دکتری آزاد سال ۹۴ به همراه پاسخ تشریحی.
 ۶. حدود ۱۸ سوال تشریحی آزمون دکتری دانشگاه آزاد اسلامی به همراه پاسخ آن‌ها
 ۷. حدود ۲۰۰ سوال تشریحی که پاسخ آن‌ها را می‌توانید از سایت انتشارات فناوری نوین به آدرس www.fanavarienovin.net دانلود کنید.
- از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هر گونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.
- در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

مؤلفین

fanavarienovin@gmail.com

آشنایی با سیستم‌های توزیع شده

در این فصل به مفاهیم سیستم‌های توزیع شده^۱ می‌پردازیم. اهداف سیستم‌های توزیع شده و انواع آن‌ها را مورد بررسی قرار می‌دهیم. در پایان، انواع پیاده‌سازی سیستم‌های توزیع شده را بیان نموده، آن‌ها را با سیستم‌های متمرکز^۲ مقایسه خواهیم کرد.

۱-۱. تعریف سیستم توزیع شده

سیستم توزیع شده، مجموعه‌ای از کامپیوترهای مستقل و ناهمگن است که از دید کاربر به صورت یک سیستم واحد (منسجم) به نظر می‌رسد. در این تعریف دو ویژگی بسیار مهم وجود دارد که عبارت‌اند از:

۱. از بعد سخت‌افزاری، ماشین‌ها مستقل و خودمختار هستند.
۲. از بعد نرم‌افزاری، کاربران فکر می‌کنند که با یک ماشین سروکار دارند. یعنی، از دید کاربر به عنوان یک سیستم واحد و منسجم به نظر می‌رسند.

یکی از مثال‌های سیستم توزیع شده، شبکه جهانی اینترنت است. در این شبکه علاوه بر این که ماشین‌ها به صورت مستقل و خودمختار می‌باشند، کامپیوترها و نرم‌افزارها می‌توانند ناهمگن باشند. از طرف دیگر، کاربر به سادگی با وارد کردن نام منحصر به فرد^۳ یا یکتا (نظیر www.fanavarienovin.net) می‌تواند به داده‌ها دست‌یابی داشته باشد، بدون این که لازم باشد، محل و آدرس واقعی داده را بداند.

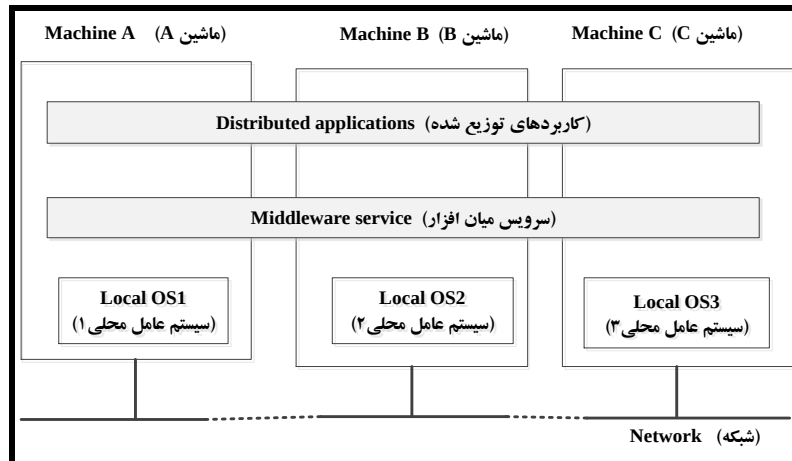
در سیستم‌های توزیع شده، هر کامپیوتر سیستم عامل خاص خودش را دارد (ممکن است نوع سیستم عامل‌ها در ماشین‌های مختلف متفاوت باشد). بنابراین، وظیفه واحد شدن (منسجم) سیستم‌ها را نمی‌توان بر عهده یک سیستم عامل قرار داد. پس سیستم‌های توزیع شده به منظور منسجم نمودن سیستم‌ها از ابزاری به نام میان‌افزار^۴ استفاده می‌کنند (شکل ۱-۱). همان‌طور که در این شکل مشاهده می‌شود، میان‌افزار بین برنامه‌های کاربردی سیستم عامل‌ها قرار می‌گیرد تا کاربر از مسائلی که در سیستم‌های عامل و سخت‌افزارها اتفاق می‌افتد، اطلاعی نداشته باشد (بدون اطلاع باشد). در واقع، میان‌افزار جهت پشتیبانی از کامپیوترها و شبکه‌های غیر همگن است تا سیستم‌های مختلف از نظر کاربر به عنوان یک سیستم واحد به نظر به رسند.

۱-۲. اهداف سیستم‌های توزیع شده

سیستم‌های توزیع شده اهداف زیر را دارند:

۱. برقراری ارتباط آسان بین کاربران و منابع (Easily Connecting Users and Resources)
۲. شفافیت (Transparency)
۳. باز بودن سیستم (Openness)
۴. مقیاس پذیری (توسعه پذیری Scalability)

^۱. Distributed Systems ^۲. Centralized Systems ^۳. URL ^۴. Middleware



شکل ۱-۱ میان افزار و سیستم های توزیع شده.

۱-۲-۱. برقراری ارتباط آسان بین منابع و کاربران

یکی از اهداف اصلی سیستم های توزیع شده اتصال آسان بین کاربران و منابع راه دور و به اشتراک گذاری این منابع بین کاربران است (البته دسترسی از راه دور و به اشتراک گذاری باید کنترل شده باشد). به اشتراک گذاری منابع دلایل زیادی دارد که عبارت اند از:

- به اشتراک گذاری منابع موجب صرفه جویی اقتصادی می شود. چون، کاربران به جای این که از چندین چاپگر استفاده کنند می توانند از یک چاپگر به صورت اشتراکی استفاده نمایند.
- امکان تبادل اطلاعات بین کاربران آسان تر خواهد شد. به عنوان مثال، تبادل اطلاعات از طریق اینترنت ساده می باشد.

از طرف دیگر، چون برخی از کارها ذاتاً به صورت توزیعی انجام می شوند (مانند شبکه بانک ها)، با اتصال منابع و کاربران این امکان فراهم می شود. البته باید دقت داشته باشید که هر چه استفاده اشتراکی از منابع افزایش می یابد، مسئله امنیت نیز مهم تر خواهد شد. یعنی، نباید اجازه داد تا افراد غیرمجاز به داده ها دسترسی داشته باشند. برای برقراری امنیت می توان علاوه بر استفاده از کلمه عبور جهت دسترسی از روش های رمزگذاری داده ها نیز استفاده نمود تا جلوی مشکلات امنیتی از قبیل استراق سمع و دزدیده شدن اطلاعات گرفته شود.

۱-۲-۲. شفافیت

یکی از اهداف مهم دیگر سیستم های توزیع شده، پنهان سازی این واقعیت است که فرآیندها^۱ و منابع به صورت فیزیکی روی چندین کامپیوتر متعدد توزیع شده اند. در واقع، کاربران نباید متوجه این واقعیت شوند که فرآیندها (کارها) به صورت توزیعی بر روی چندین ماشین انجام می شوند. این مسئله شفافیت نام دارد. سیستم توزیع شده ای که بتواند این واقعیت را مخفی کند و خود را به کاربر و برنامه های کاربردی اش، طوری نشان دهد که مانند یک کامپیوتر واحد به نظر آید، یک سیستم شفاف^۲ نامیده می شود. شفافیت انواع مختلف دارد که عبارت اند از:

^۱.Process ^۲.Transparence System

۱. شفافیت دسترسی^۱، همان‌طور که بیان گردید کامپیوترها در سیستم‌های توزیع‌شده می‌توانند ناهمگن باشند. این امر موجب می‌شود تا نمایش داده‌ها در کامپیوترهای مختلف متفاوت باشد. به‌عنوان مثال، کامپیوترهای نوع اینتل از فرمت (Little endian) برای تبادل داده استفاده می‌کنند. یعنی، اول بایت باارزش، سپس بایت کم‌ارزش انتقال می‌یابد. اما، کامپیوترها نوع Sun SPARC از فرمت (Big endian) برای تبادل اطلاعات استفاده می‌کنند (یعنی، ابتدا بایت کم‌ارزش، سپس بایت با ارزش انتقال می‌یابد). از طرف دیگر، چون منابع و کاربران با هم فاصله دارند، کاربران باید بتوانند از راه دور به منابع متصل شوند. شفافیت دسترسی (دستیابی)، تفاوت در نمایش داده‌ها و چگونگی دستیابی به منابع را از دید کاربران مخفی می‌نماید. بنابراین، سیستمی که شفافیت دسترسی را دارد، از دید کاربران تفاوتی در روش دسترسی به داده محلی و راه دور وجود ندارد و از دید کاربران چگونگی تبدیل و انتقال داده‌های مختلف روی کامپیوترهای متفاوت پنهان است.

۲. شفافیت مکانی^۲، مکان فیزیکی منابع را از دید کاربران مخفی می‌کند. یعنی، کاربران بدون این‌که محل واقعی منابع را بدانند می‌توانند به آن‌ها دسترسی داشته باشند. به‌عنوان مثال، آدرس www.google.com را در نظر بگیرید. کاربران بدون این‌که آدرس واقعی آن را بدانند، به داده‌های آن دسترسی دارند. در این ساختار معمولاً آدرس‌ها و نام‌های منابع منطقی هستند. در فصل پنجم با انواع نام‌گذاری منابع در سیستم‌های توزیع‌شده آشنا خواهیم شد.

۳. شفافیت مهاجرت^۳، انتقال منابع از یک مکان به مکان دیگر را از دید کاربر مخفی می‌نماید. به‌عنوان مثال، سرویس‌دهنده^۴ یا هو که در آمریکا قرار دارد، اگر به کانادا انتقال یابد، این انتقال از دید کاربر پنهان می‌ماند.

۴. شفافیت جابه‌جایی^۵، انتقال منابع در حال استفاده از یک مکان به مکان دیگر را از دید کاربران پنهان می‌نماید. نمونه‌ای از شفافیت جابه‌جایی، استفاده کاربران سیار از لپ‌تاپ^۶ در حال انتقال از محلی به محل دیگر است. یعنی، این کاربران بدون قطع اتصال از طریق اتصال بی‌سیم می‌توانند از نقطه‌ای به نقطه دیگر منتقل شوند. فرق بین شفافیت مهاجرت و شفافیت جابه‌جایی این است که در شفافیت جابه‌جایی منبع در حال استفاده جابه‌جا می‌شود. ایجاد شفافیت جابه‌جایی از ایجاد شفافیت مهاجرت پیچیده‌تر است.

۵. شفافیت تکرار (تکثیر)^۷، معمولاً چندین نسخه تکراری از یک داده یا منبع در سیستم‌های توزیع‌شده استفاده می‌شود تا قابلیت اعتماد و قابلیت دسترسی افزایش یابد. شفافیت تکرار، وجود چندین نسخه از داده یا منبع را از دید کاربران مخفی می‌کند و کاربران مانند زمانی عمل می‌کنند که تنها یک نسخه در سیستم وجود داشته باشد. قاعدتاً، سیستمی که شفافیت تکثیر را دارد باید شفافیت مکانی را نیز داشته باشد تا کاربران از محل واقعی داده‌ها بی‌خبر باشند. برای این منظور، باید تمام منابع تکثیرشده یک نام داشته باشند. زیرا، در غیر این صورت کاربران نمی‌توانند به کامپیوترهای مکان دیگر مراجعه کنند.

^۱.Access Transparency ^۲.Location Transparency ^۳.Migration Transparency ^۴.Server
^۵.Relocation Transparency ^۶.Laptop ^۷.Replication Transparency

۶. شفافیت هم‌روندی^۱، یکی از اهداف سیستم‌های توزیع شده امکان استفاده مشترک و هم‌زمان منابع توسط کاربران است تا صرفه‌جویی اقتصادی حاصل شود. در این صورت، کاربران می‌توانند برای اخذ منابع با هم رقابت کنند. شفافیت هم‌روندی، این واقعیت را که منابع به‌صورت اشتراکی و هم‌زمان مورد استفاده قرار می‌گیرند و کاربران برای اخذ منابع با یکدیگر رقابت می‌کنند را از دید کاربران پنهان می‌کند. به عبارت دیگر، هر کاربر مانند این عمل می‌کند که منبع فقط در اختیار خودش است و سیستم موظف است که با قوانین خاصی داده‌ها را به‌روز (به هنگام) کرده و داده‌های صحیح را در اختیار کاربران دیگر قرار دهد. دو روش برای دستیابی به شفافیت هم‌روندی وجود دارد که عبارت‌اند از: ۱. روش قفل‌گذاری و دسترسی انحصاری به منابع ۲. استفاده از مفهوم تراکنش. در فصل هفتم روش‌های تکثیر داده را می‌آموزیم.

۷. شفافیت خرابی^۲، ترمیم خرابی منابع را از دید کاربران مخفی می‌کند. یعنی، اگر سیستمی خراب شود، کاربر متوجه نشود که از طریق تکثیر (تکرار) قابل حل است. به عبارت دیگر، با خراب شدن یک منبع سیستم این قابلیت را داشته باشد تا با بقیه منابع موجود خرابی را پوشش دهد. مشکل عمده در شفافیت خرابی، عدم توانایی در تمایز بین منابع خراب (غیرفعال) و منابع فعال بسیار کند است. به عنوان مثال، زمانی که تماس به سرویس‌دهنده وب که ترافیک زیادی دارد، می‌رسد ممکن است فرصت زمانی (مهلت) مرورگر خاتمه یابد. در این صورت، گزارش می‌دهد که صفحه وب وجود ندارد. در اینجا کاربر نمی‌تواند نتیجه بگیرد که سرویس‌دهنده واقعاً خراب شده است یا به دلیل ترافیک، فرآیندها قادر به پاسخ‌گویی در فرصت زمانی تعیین شده نیست.

۸. شفافیت ماندگاری (پایداری)^۳، همان‌طور که می‌دانید داده‌ها در یک کامپیوتر بر روی حافظه قرار می‌گیرند. حافظه‌ها دو نوع‌اند:

➤ حافظه اصلی، داده‌های ناپایدار را نگهداری می‌کند که با قطع جریان برق محتوی حافظه پاک می‌شود.
➤ حافظه جانبی، داده‌های پایدار (مانند گار) را نگهداری می‌کند. این داده‌ها با قطع جریان برق پاک نمی‌شوند.

۹. شفافیت پایداری، مکان ذخیره داده را از دید کاربر مخفی می‌نماید. به عبارت دیگر، کاربر کاری ندارد که داده کجا ذخیره می‌شود، بلکه سیستم تصمیم می‌گیرد که چه موقع داده‌ای در حافظه اصلی باشد یا در حافظه جانبی. به عنوان مثال، در بانک اطلاعاتی شی گرا^۴ زمانی که متدی از یک شی فراخوانی می‌شود باید این شی از حافظه جانبی به حافظه اصلی کپی گردد و این فرآیند کپی باید از دید کاربر پنهان بماند.

۳-۲-۱. باز بودن

سومین هدف سیستم‌های توزیع شده، باز بودن^۵ است. یک سیستم توزیع شده باز، سیستمی است که سرویس‌ها را بر اساس قواعد و قوانین استاندارد ارائه می‌دهد که قواعد نحو^۶ و معنایی آن سرویس‌ها، در سیستم‌های توزیع شده توصیف می‌گردد. یعنی، اگر بخواهیم نیازها یا سرویس‌های جدید ارائه کنیم، میان‌افزار

^۱.Concurrency Transparency ^۲.Failure Transparency ^۳.Recovery ^۴.Persistence Transparency
^۵.Object Oriented Database ^۶.Object ^۷.Openness ^۸.Syntax

باید بتواند سرویس‌های جدید را ارائه دهد. لازمه این کار این است که یک زبان تعریف واسط (IDL)^۱ نظیر جاوا برای میان‌افزار استفاده شود. در IDL پیش‌تر به توصیف توابع قابل دسترسی، پارامترهای آن‌ها، حالت‌های استثنایی که ممکن است، اتفاق بی افتد و غیره پرداخته می‌شود. به عبارت دیگر، IDL اجازه می‌دهد که یک فرآیند دلخواه که به یک واسط نیاز دارد تا با فرآیند دیگری ارتباط برقرار کند، این واسط را تولید کند. در این صورت، برای یک برنامه کاربردی که نوشته می‌شود مهم نیست که بر روی سیستم عامل لینوکس اجرا می‌شود یا بر روی سیستم عامل ویندوز. جهت پشتیبانی از باز بودن باید تمام تعریف‌ها کامل و فارغ از پیاده‌سازی باشند. کامل بودن خیلی مشکل است. چون، طراحی یکسری مفروضاتی دارد و فردی که می‌خواهد سیستم را توسعه دهد ممکن است مفروضات دیگر داشته باشد. برای این که تعریف‌ها کامل و فارغ از پیاده‌سازی باشند باید ویژگی‌های زیر را داشته باشند:

۱. قابلیت انعطاف^۲، یعنی بتوان در سیستم از قطعات مختلف از سازندگان متفاوت استفاده کرد. همچنین بتوان به سیستم قطعه‌ای را اضافه کرد یا قطعه موجود را تعویض نمود. این عمل نباید در کار قطعات در حال کار اختلالی ایجاد نماید. به عبارت دیگر، سیستم باز باید، بسط پذیر^۳ باشد.
۲. قابلیت حمل^۴، بدون این که تغییری در برنامه کاربردی ایجاد گردد، بر روی سیستم توزیع‌شده دیگری اجرا نمود (ممکن است سیستم‌های توزیع‌شده‌ای که برنامه کاربردی در آن نصب می‌شود و بر روی آن اجرا می‌شود، متفاوت و ناهمگن باشند).

۴-۲-۱. توسعه‌پذیری

- یکی از اهداف بسیار مهم سیستم‌های توزیع‌شده، توسعه‌پذیری (مقیاس‌پذیری)^۵ است. یعنی، به راحتی بتوان سیستم را توسعه و گسترش داد. توسعه‌پذیری را می‌توان در سه بعد بررسی نمود که عبارت‌اند از:
۱. توسعه‌پذیری از لحاظ اندازه^۶، یعنی به راحتی بتوان تعداد منابع و کاربران سیستم را افزایش داد.
 ۲. توسعه‌پذیری از لحاظ جغرافیایی^۷، یعنی بتوان فاصله فیزیکی کاربران و منابع را افزایش داد و کاربران و منابع از هم دور باشند.
 ۳. توسعه‌پذیری از لحاظ سرپرستی^۸، یعنی سیستم با توجه به داشتن تعداد زیادی سازمان‌های اجرایی مستقل، باز هم به سادگی قابل اداره باشد.
- متأسفانه با توسعه سیستم در یک یا چند بعد معمولاً کارایی آن کاهش می‌یابد. توسعه دادن سیستم مشکلات زیادی را ایجاد می‌نماید که برخی از آن‌ها در زیر آمده‌اند:
- به عنوان مثال، در توسعه اندازه، اگر کاربران یا منابع زیادی را سرویس‌دهی کنیم با محدودیت‌های زیر مواجه خواهیم شد:
۱. محدودیت‌های سرویس‌های متمرکز^۹، اگر سرویس‌ها به صورت متمرکز روی یک سرویس‌دهنده قرار گیرند، با افزایش کاربران و منابع، این سرویس‌دهنده به یک گلوگاه^{۱۰} تبدیل خواهد شد که باعث افزایش ترافیک و کاهش کارایی می‌شود.

^۱.Interface Definition Language ^۲.Flexibility ^۳.Extensible ^۴.Portable ^۵.Scalability
^۶.Size ^۷.Geographical ^۸.Administrative ^۹.Centralized Services ^{۱۰}.Centralized algorithms

۲. **محدودیت داده‌های متمرکز**^۱، اگر داده‌ها به صورت متمرکز باشد، توسعه آن مشکل می‌باشد. به عنوان مثال، برای سازمان بزرگی مانند ثبت احوال قرار دادن تمام اطلاعات در یک جدول موجب ایجاد مشکل در توسعه می‌گردد. برای حل محدودیت‌هایی از قبیل سرویس‌ها و داده‌های متمرکز می‌توان از **تکرار** استفاده نمود. تکرار و تکثیر را در فصل هفتم می‌آموزیم.

۳. **محدودیت الگوریتم‌های متمرکز**^۲، الگوریتم‌های بهینه مسیریابی، تصمیم‌گیری را بر اساس اطلاعات تمام ماشین‌ها انجام می‌دهند. یعنی، یک ماشین اطلاعات سراسری را از تمام ماشین‌ها تحویل گرفته، عمل مسیریابی را انجام می‌دهد. در سیستم‌های توزیع شده بزرگ، تعداد زیادی پیام باید مسیریابی شوند. چون، این پیام‌ها برای مسیریابی باید به یک ماشین انتقال یابند، سربار زیادی را به سیستم تحمیل خواهند کرد. پس، برای بهبود توسعه‌پذیری باید از الگوریتم‌های نامتمرکز استفاده نمود. الگوریتم‌های نامتمرکز ویژگی‌های زیر را دارند:

🌟 هیچ ماشینی اطلاعات کاملی در مورد وضعیت سیستم ندارد.

🌟 ماشین‌ها فقط بر اساس ساعت محلی‌شان تصمیم می‌گیرند.

🌟 خرابی یک ماشین موجب خرابی الگوریتم نمی‌گردد.

🌟 هیچ تصویری برای وجود یک ساعت سراسری در سیستم وجود ندارد.

سه مورد اول، نیاز به توضیح ندارند. اما، مورد چهارم نیاز به توضیح دارد. چون در سیستم توزیع شده هیچ‌گاه نمی‌توان مطمئن شد که تمامی کامپیوترها (ماشین‌ها) ساعت یکسانی دارند. بنابراین، الگوریتم‌ها نباید بر اساس ساعت مشترک کار کنند. چون هر چه اندازه سیستم بزرگ‌تر می‌شود، احتمال اختلاف بین ساعت‌ها نیز بیش‌تر خواهد شد.

توسعه جغرافیایی نیز مشکلاتی دارد که عبارت‌اند از:

۱. **مهم‌ترین مشکلی** که در توسعه جغرافیایی یک سیستم توزیع شده اتفاق می‌افتد، تأخیر رسیدن بسته‌های اطلاعاتی زیادتر می‌شود. بنابراین، قابلیت اعتماد به صحت بسته‌های ارسالی پایین می‌آید.

۲. علاوه بر این فعلاً نمی‌توان سیستم‌های توزیع شده‌ای را که برای شبکه محلی طراحی شده‌اند، توسعه داد. زیرا، این شبکه‌ها مبتنی بر ارتباط **همکن** هستند. در این نوع ارتباط سرویس‌گیرنده سرویسی را درخواست کرده، تا زمانی که پاسخ را دریافت نکند متوقف می‌شود. از این روش نمی‌توان در سیستم‌های توزیع شده استفاده کرد. چون بسیار کند است.

مشکل اصلی توسعه سرپرستی تداخل سیاست‌های سازمان‌های اجرایی مختلف در سیستم است. هر چه تعداد سازمان‌های اجرایی مستقل بیش‌تری در سیستم وجود داشته باشد، احتمال به‌روز تداخل بین آن‌ها زیادتر خواهد شد.

۳-۱. تکنیک‌های توسعه‌پذیری

همان‌طور که بیان گردید، توسعه‌پذیری سیستم‌های توزیع شده از لحاظ اندازه، جغرافیایی و سرپرستی مشکلات و محدودیت‌های را ایجاد می‌کند. برای بهبود مشکلات توسعه‌پذیری سیستم توزیع شده سه تکنیک وجود دارد که عبارت‌اند از:

^۱.Centralized Data ^۲.Centralized algorithms

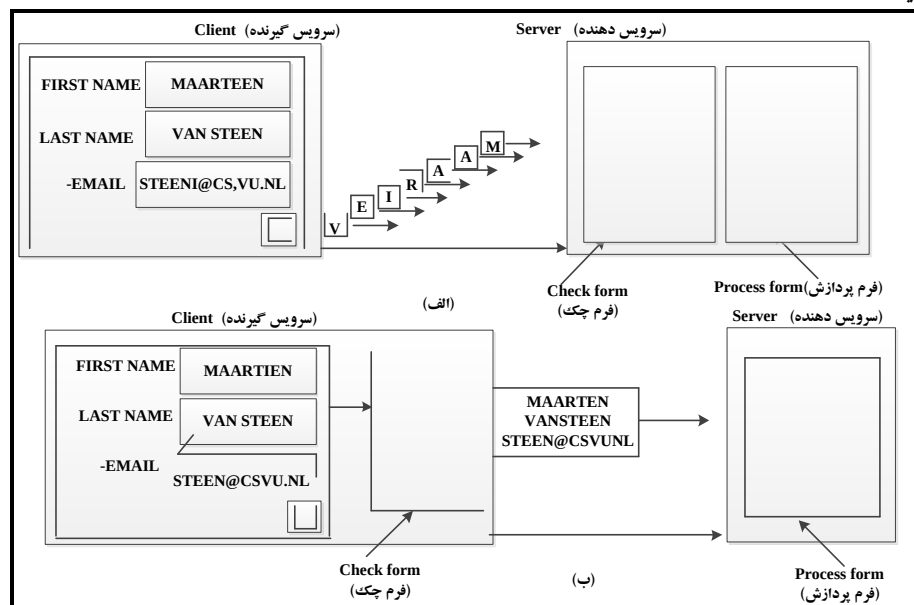
۱. مخفی کردن تأخیرهای ارتباطات (Hiding Communication Latencies)

۲. توزیع (Distributed) ۳. تکثیر یا تکرار (Replication)

۱-۳-۱. مخفی کردن تأخیر ارتباطات

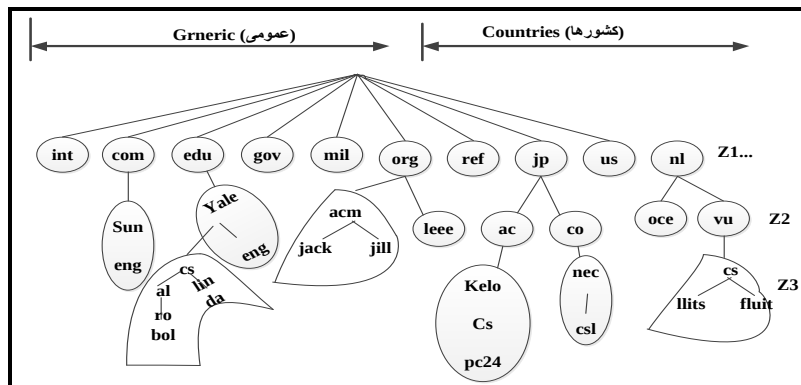
همان‌طور که بیان گردید، راه رسیدن به قابلیت توسعه جغرافیایی مناسب، پنهان کردن تأخیر ارتباطات است. یعنی، برنامه‌های کاربردی باید طوری نوشته شوند که فقط از ارتباطات آسنکرون^۱ (غیرهم‌زمان) استفاده کنند. در ارتباط غیر هم‌زمان، سرویس‌گیرنده درخواستی را به سرویس‌دهنده می‌فرستد، تا زمانی که درخواست پاسخ داده شود، به کارش ادامه می‌دهد و اعمال دیگر را انجام می‌دهد. وقتی یک پاسخ به سرویس‌گیرنده (درخواست‌کننده) می‌رسد، در کار آن وقفه ایجاد می‌شود و عملیات مربوط به درخواست قبلی کامل می‌شود. ارتباط غیر هم‌زمان در سیستم‌های پردازش-دسته‌ای^۲ و کاربردهای پردازش موازی^۳ مفید است. چون، در این کاربردها، تعدادی عمل که کم‌ویش مستقل هستند، انجام می‌شوند و زمانی که در انتظار پاسخ سرویس‌دهنده است، کاربردهای دیگر می‌توانند به کارشان ادامه دهند.

نمونه‌ای از پنهان کردن تأخیر ارتباط در شکل ۲-۱ آمده است. در بخش (الف) شکل، وقتی سرویس‌گیرنده درخواستی را از سرویس‌دهنده دارد، مدت‌زمان نسبتاً زیادی را باید منتظر پاسخ بماند. چون عمل کنترل فرم (Check form) سرویس‌دهنده انجام می‌شود و از طرف دیگر، تعداد سرویس‌گیرنده‌ها نیز زیاد است. در صورتی که همان‌طور که در بخش (ب) شکل ۲-۱ می‌بینید، با قرار دادن عمل کنترل فرم (Check form) به عهده سرویس‌گیرنده‌ها بار سرویس‌دهنده‌ها کاهش می‌یابد و زمان پاسخ گوئی نیز کاهش خواهد یافت.



شکل ۲-۱. تفاوت بین تکمیل و کنترل فرم توسط سرویس‌دهنده (الف) و سرویس‌گیرنده (ب).

^۱. Asynchronous Communication ^۲.Batch Processing System ^۳.Parallel Processing



شکل ۳-۱ تقسیم فضای نام DNS به مناطق.

۲-۳-۱. توزیع

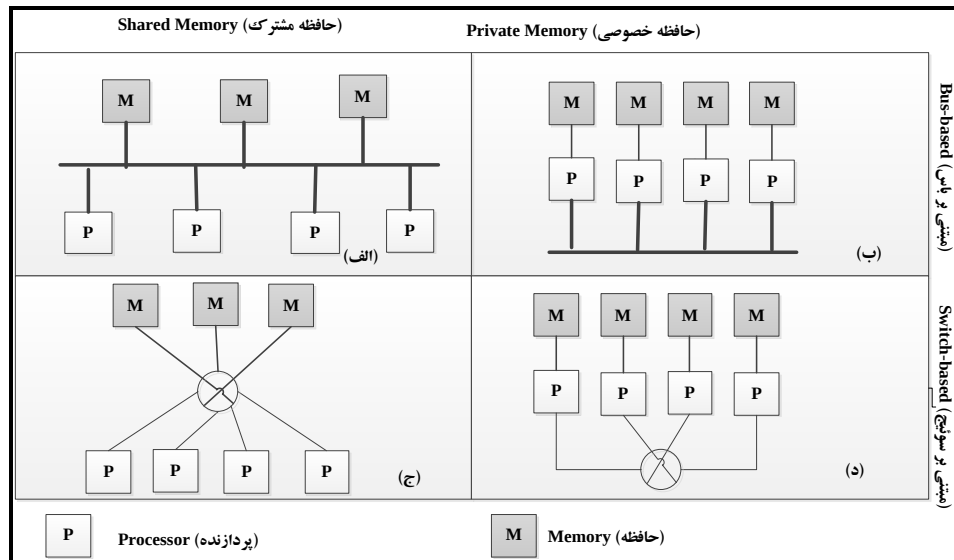
توزیع، یکی دیگر از تکنیک‌های توسعه‌پذیری است. توزیع در حقیقت تبدیل یک قطعه^۱ به قطعات کوچک‌تر و پخش کردن آن‌ها در سیستم است. سیستم نام‌گذاری دامنه (DNS)^۲ در اینترنت مثال خوبی از توزیع است. DNS، برای تبدیل نام‌های منطقی اینترنتی (URL) به آدرس‌های واقعی آن است. این سیستم در اینترنت به صورت درختی و سلسله‌مراتبی است (همان‌طور که در شکل ۳-۱ می‌بینید). در این سیستم شبکه به یک سری مناطق تقسیم‌شده و کاربران از این سیستم توزیع‌شده DNS استفاده می‌کنند.

۳-۳-۱. تکثیر

تکثیر، سومین تکنیک برای افزایش توسعه‌پذیری است. در تکثیر، قطعات برنامه در سیستم توزیع‌شده تکرار می‌شوند. تکثیر نه تنها قابلیت دسترسی را افزایش می‌دهد، بلکه به توزیع متوازن بار بین سیستم‌ها نیز کمک می‌کند تا کارایی بهبود یابد. همچنین در توسعه جغرافیایی سیستم، وجود یک نسخه تکراری از داده موردنظر در نزدیکی مقصد باعث کاهش تأخیر در ارتباط خواهد شد. نوع خاصی از تکثیر استفاده از حافظه نهان^۳ است. حافظه نهان، یک کپی از کل یا بخشی از داده موردنظر در سرویس‌گیرنده^۴ است که به طور موقت در دسترس آن قرار می‌گیرد تا کارایی افزایش یابد. تفاوت عمده بین تکثیر و حافظه نهان این است که تکثیر یک منبع تضمینی است که توسط سرویس‌دهنده و مالک منبع گرفته می‌شود. ولی، استفاده از حافظه نهان توسط سرویس‌گیرنده انجام می‌شود و مالک در آن نقشی ندارد. از آنجایی که کپی‌ها متعددی از یک منبع وجود دارد، تغییر در یک کپی باعث می‌شود تا با بقیه فرق کند، در نتیجه تکثیر موجب عدم سازگاری^۵ داده می‌شود. مسئله سازگاری را در فصل ۷ می‌آموزیم.

مزایای تکثیر عبارت‌اند از: ۱. در دسترس بودن را بالا می‌برد. ۲. توازن بار ایجاد می‌کند. ۳. زمان سرویس‌دهی را کاهش می‌دهد. ۴. کارایی را افزایش می‌دهد. مشکلات کش کردن و تکثیر عبارت‌اند از: ۱. سازگاری (Consistency) ۲. مشکل هم‌زمان‌سازی ساعت‌ها ۳. امنیت

^۱. Component ^۲. Domain Name System ^۳. Cache ^۴. Client ^۵. Consistency



شکل ۴-۱ انواع سیستم‌های چندپردازنده‌ای.

۴-۱. سیستم‌های چندپردازنده‌ای

سیستم چندپردازنده‌ای^۱، از یک کامپیوتری که دارای چندین پردازنده (CUP) و یک حافظه مشترک می‌باشد، تشکیل شده است (حافظه اشتراکی همان RAM می‌باشد). انواع این سیستم‌ها در شکل ۴-۱ نشان داده شده‌اند. از جمله ویژگی این سیستم‌ها عبارت‌اند از:

- برای سرویس خاصی طراحی شده‌اند.
- یکی از سرویس‌های سیستم‌های چند کامپیوتری^۲ را انجام می‌دهند (سیستم‌های چند کامپیوتری را در ادامه می‌بینید).
- هدف این سیستم‌ها افزایش کارایی است.
- این سیستم‌ها فضای حافظه مشترک دارند.
- این سیستم‌ها اطلاعات را از طریق حافظه مشترک تبادل می‌کنند.
- این سیستم‌ها همگن هستند. یعنی، از یک نوع‌اند و ساعت‌شان (کلاک‌شان) هم یکسان است.

سیستم‌های چندپردازنده‌ای به دو دسته زیر تقسیم می‌گردند:

- سیستم‌های با دسترسی یکنواخت به حافظه اشتراکی (UMA^۳).
- سیستم‌های با دسترسی غیریکنواخت به حافظه اشتراکی (NUMA^۴).

۴-۱-۱. سیستم‌های چندپردازنده‌ای UMA با سوئیچ

دو روش پیاده‌سازی چندپردازنده‌ای UMA با سوئیچ عبارت‌اند از (شکل ۵-۱):

^۱.Multi Processor

^۲.Multi Computer

^۳.Uniform Memory Access

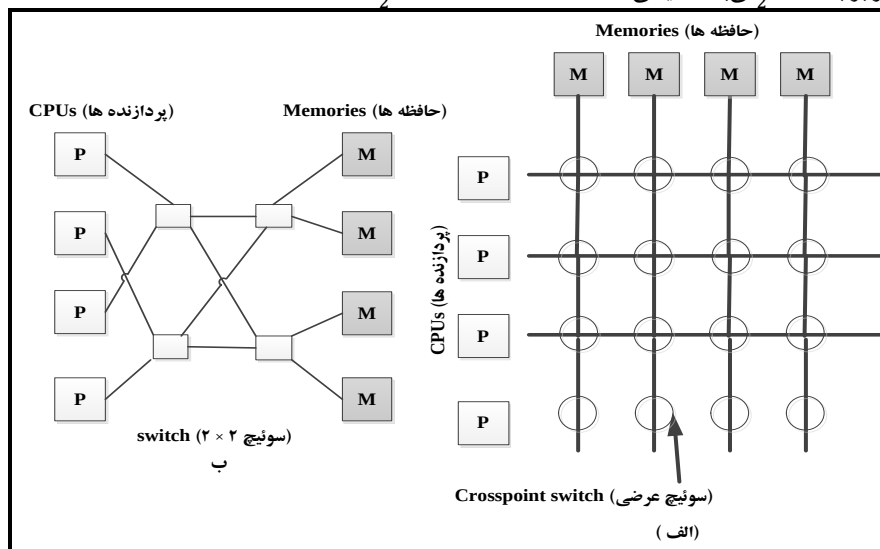
^۴. None Uniform Memory Access

۱. سیستم‌های چندپردازنده‌ای UMA با سوئیچ عرضی^۱، در این نوع سیستم‌ها، تمام پردازنده‌ها به صورت ماتریسی از طریق سوئیچ‌ها به هم متصل هستند. در این روش هر پردازنده در هر لحظه می‌تواند به یکی از حافظه‌ها دسترسی داشته باشد (به طور هم‌زمان دو پردازنده نمی‌توانند یک حافظه را در اختیار بگیرند). مزیت این روش این است که با افزایش دادن تعداد گذرگاه‌ها ترافیک را کاهش داده است. همچنین می‌توان با موازی‌سازی درخواست پردازنده‌ها از گذرگاه برای دسترسی به حافظه استفاده نمود. عیب این روش، تعداد زیاد سوئیچ و گرانی شبکه است. زیرا، اگر n پردازنده و n حافظه در شبکه موجود باشد، به n^2 سوئیچ نیاز است.

این روش عموماً مربعی است (می‌تواند مربعی هم نباشد).

۲. سیستم‌های چندپردازنده‌ای UMA با استفاده از شبکه سوئیچ امگا^۲، همان‌طوری که در روش قبلی بیان گردید، با افزایش تعداد پردازنده‌ها و حافظه‌ها، تعداد سوئیچ‌ها خیلی افزایش می‌یابد. برای کاهش تعداد سوئیچ‌ها از این روش استفاده می‌شود. در شکل ۵-۱ یک سوئیچ 2×2 را می‌بینید که به دو حالت مستقیم و ضربدری ورودی خروجی را به هم وصل می‌کند. این روش سوئیچ چندمرحله‌ای نام دارد. اگر تعداد پردازنده‌ها و حافظه n باشد، $\frac{n}{2}$ سوئیچ در هر مرحله وجود دارد. بنابراین، تعداد سوئیچ برابر با $\frac{n}{2} \log_2^n$ است. مثال: اگر n (تعداد پردازنده‌ها) برابر ۱۰۲۴ باشد، تعداد سوئیچ‌ها را به روش‌های سوئیچ عرضی و سوئیچ امگا به دست آورید:

در روش سوئیچ عرضی تعداد سوئیچ برابر با n^2 است (یعنی $(1024)^2$). اما، در روش سوئیچ امگا، تعداد سوئیچ برابر با $\frac{n}{2} \log_2^n$ می‌باشد (یعنی، $\frac{1024}{2} \times \log_2^{1024} = 5120$).



شکل ۵-۱ مدل‌های سیستم‌های چندپردازنده‌ای UMA.

^۱.Crossbar Switch ^۲.Omega Switch

۵-۱. سیستم‌های با دسترسی یکنواخت به حافظه اشتراکی

در این سیستم‌ها چندپردازنده به یک حافظه دسترسی دارند. این سیستم‌ها به دو روش پیاده‌سازی شده‌اند:

۱. سیستم‌های چندپردازنده‌ای UMA مبتنی بر گذرگاه (BUS)، این سیستم‌ها از چندپردازنده مستقل و یک حافظه مشترک برای تمام پردازنده‌ها تشکیل می‌شود (شکل ۶-۱) که پردازنده‌ها، داده‌ها و اطلاعات مورد نیاز خود را از طریق گذرگاه از حافظه مشترک دریافت می‌کنند. مشکل اصلی این روش، این است که به دلیل ارتباط زیاد بین پردازنده‌ها و حافظه اشتراکی ترافیک گذرگاه افزایش یافته و موجب کاهش کارایی می‌شود. برای رفع این مشکل دو روش پیشنهاد شده است عبارت‌اند از:

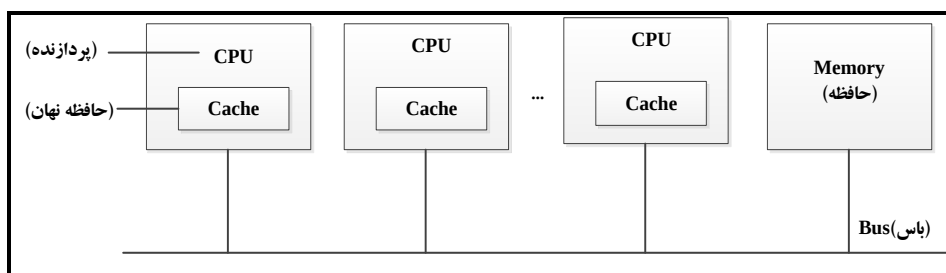
➤ هر پردازنده دارای یک حافظه نهان^۱ نیز باشد. در این مدل سیستم‌ها از چندپردازنده مستقل و یک حافظه اشتراکی برای تمام پردازنده تشکیل شده‌اند که داده‌های خود را در حافظه اشتراکی دریافت می‌کنند (شکل ۶-۱).

۲. سیستم‌های چندپردازنده‌ای با دسترسی غیریکنواخت به حافظه اشتراکی (NUMA). این سیستم‌ها دارای مشخصات زیر هستند:

- یک حافظه با آدرس منحصر به فرد برای تمامی پردازنده‌ها وجود دارد.
 - دسترسی به حافظه‌های راه دور توسط فرمان‌های Load (بار کردن) و Store (بازیابی) انجام می‌شود.
 - دسترسی به حافظه راه دور کندتر از حافظه محلی است.
 - در این روش الگوریتم‌های پیچیده‌تری نیاز است.
- همان‌طور که مشاهده کردید، نتیجه می‌گیریم که داشتن سیستم‌های چندپردازنده‌ای ممکن است، اما مشکل و گران می‌باشد.

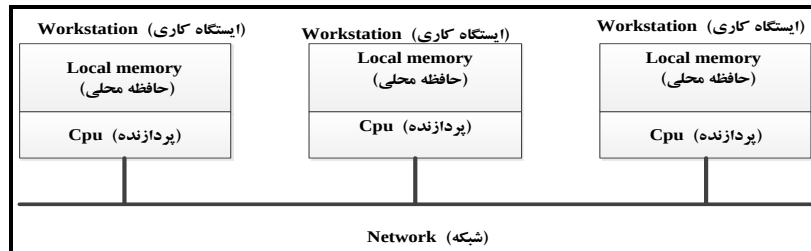
۶-۱. سیستم‌های چند کامپیوتری

در این سیستم‌ها برخلاف سیستم‌های چندپردازنده‌ای که پردازنده‌ها از یک حافظه مشترک استفاده می‌کنند، هر یک از پردازنده‌ها حافظه مستقل به خود را دارند. سیستم‌های چند کامپیوتری^۲ نیز به دو روش زیر پیاده‌سازی می‌شوند:



شکل ۶-۱ سیستم چندپردازنده‌ای با حافظه اشتراکی.

^۱.Cash ^۲.Multicomputer



شکل ۷-۱ سیستم چند کامپیوتری مبتنی بر گذرگاه باس (خطی).

۱. سیستم‌های چند کامپیوتری مبتنی بر گذرگاه^۱، از آنجای که هر پردازنده حافظه محلی خودش را دارد، بنابراین ساخت سیستم‌های چند کامپیوتری (بدون حافظه مشترک) آسان است. تنها مسئله این است که پردازنده‌ها را چگونه به هم ارتباط دهیم. در شکل ۷-۱ سیستم چند کامپیوتری مبتنی بر گذرگاه را می‌بینید که توپولوژی آن مانند سیستم چندپردازنده‌ای مبتنی بر گذرگاه است. تنها تفاوت این روش با سیستم‌های چندپردازنده‌ای مبتنی بر گذرگاه این است که در این روش ترافیک بسیار کم است. علاوه بر حافظه اشتراکی هر پردازنده دارای یک حافظه نهان نیز می‌باشد. وجود حافظه نهان موجب خواهد شد تا اطلاعات بیشتری در پردازنده بافر شده و تعداد دفعات مراجعه به حافظه اشتراکی کاهش یابد. بنابراین ترافیک روی گذرگاه کم‌تر می‌گردد.

۲. تخصیص حافظه اختصاصی به هر پردازنده، در این مدل، علاوه بر حافظه مشترک بین پردازنده‌ها، هر پردازنده یک حافظه اختصاصی نیز دارد که اعمالی از قبیل پشته‌ها، برنامه و جدول در این حافظه قرار می‌گیرد.

۷-۱. سیستم‌های چند کامپیوتری مبتنی بر سوئیچ

این روش نیز مانند سیستم‌های چندپردازنده‌ای مبتنی بر سوئیچ پیاده‌سازی می‌شود. در شکل ۸-۱ دو نوع پیاده‌سازی با توپولوژی مشبک^۲ و هایپرکاب^۳ را می‌بینید که کار در توپولوژی مشبک راحت‌تر است. چون بر روی یک تخته مدار چاپی (برد) طراحی می‌شود و برای کارهایی نظیر چشم‌های ربات، آنالیز تصاویر و غیره که ذاتاً دوبعدی هستند، مناسب می‌باشد. توپولوژی هایپرکاب، یک مکعب n بعدی است (در شکل ۸-۱ چهاربعدی است که از دو مکعب تشکیل شده است). هر یک از مکعب‌ها ۸ گره^۴ و ۱۲ لبه^۵ دارند. هر گره یک پردازنده می‌باشد و هر لبه ارتباط بین دو پردازنده است. لبه‌های متناظر هر دو مکعب به هم دیگر متصل هستند که یک بعد دیگر اضافه می‌شود. به‌طور کلی، اگر این توپولوژی n بعد داشته باشد، هر پردازنده با n اتصال به پردازنده‌های دیگر ارتباط دارد.

علاوه بر حافظه مشترک هر پردازنده یک حافظه نهان نیز دارد. وجود حافظه نهان موجب خواهد شد تا اطلاعات بیش‌تری در پردازنده بافر شده و تعداد دفعات مراجعه به حافظه مشترک کاهش یابد. بنابراین ترافیک روی گذرگاه کم‌تر می‌گردد.

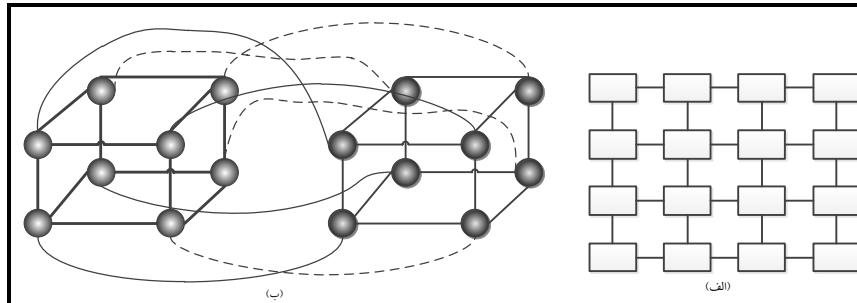
^۱.Bus Based

^۲.Grid

^۳.Hyper cube

^۴.Node

^۵.Edge



شکل ۸-۱ انواع مدل‌های ساخت چند کامپیوتری.

سیستم‌های چند کامپیوتری نسبت به چندپردازنده‌ای مزایای دارند که عبارت‌اند از:

- **سادگی توسعه‌پذیری**، در سیستم‌های چند کامپیوتری به راحتی می‌توان یک کامپیوتر را اضافه نمود، اما در سیستم‌های چندپردازنده‌ای یا این کار امکان‌پذیر نیست یا به سختی امکان‌پذیر است.
- **افزایش تحمل خطا**، تحمل خطای سیستم‌های چند کامپیوتری بیش‌تر از سیستم‌های چندپردازنده‌ای است. زیرا، اگر یکی خراب شود، مشکلی ایجاد نمی‌شود. ولی، در سیستم‌های چندپردازنده‌ای با خراب شدن یک پردازنده برای سیستم مشکل ایجاد خواهد شد.
- **افزایش خاصیت توزیعی**، در سیستم‌های چند کامپیوتری خاصیت توزیعی افزایش می‌یابد. زیرا، به عنوان مثال، اگر کامپیوتری برای پردازش تصاویر مناسب باشد، کار پردازش تصاویر به آن محول می‌گردد. اما، چون سیستم‌های چندپردازنده‌ای، پردازنده‌ها یک سرویس را ارائه می‌دهند، خاصیت توزیعی کاهش می‌یابد.
- **افزایش نسبت سرعت به هزینه**، در سیستم‌های چند کامپیوتری این پارامتر نسبت به سیستم‌های چندپردازنده‌ای بیش‌تر می‌باشد.

۸-۱. حافظه توزیع‌شده اشتراکی

برای پیاده‌سازی حافظه توزیع‌شده اشتراکی (DSM)^۱، از مفهوم **حافظه مجازی**^۲ استفاده می‌شود. با این تفاوت که به جای دیسک از RAM راه دور^۳ استفاده می‌شود. در این نوع سیستم‌ها فضای آدرس به صفحاتی (معمولاً به اندازه 4k یا 8k) تقسیم می‌شوند. این صفحات در حافظه پردازنده‌های مختلف پراکنده می‌شوند که امکان دسترسی به صفحات محلی یک پردازنده برای دیگری وجود دارد. دو تکنیک برای انجام این عمل وجود دارد:

۱. **تعریف وقفه‌های نرم‌افزاری** که به کمک آن‌ها امکان دسترسی یک پردازنده به حافظه پردازنده دیگر را فراهم می‌کند. صدور وقفه نرم‌افزاری معمولاً انتقال یک صفحه^۵ از فضای مجازی یک پردازنده به پردازنده دیگر را فراهم می‌کند.
۲. **تکرار (تکثیر) صفحه**^۶، در این حالت به جای انتقال یک صفحه به پردازنده دیگر، کپی خواهد شد. در این مورد باید بحث **انسجام**^۷ و **سازگاری**^۸ مورد توجه قرار گیرد. در شکل ۹-۱ یک فضای آدرس عمومی ۱۶

^۱.Distributed Shared Memory

^۲.Virtual Memory

^۳.Remote

^۴.Traps

^۵.Page Moving

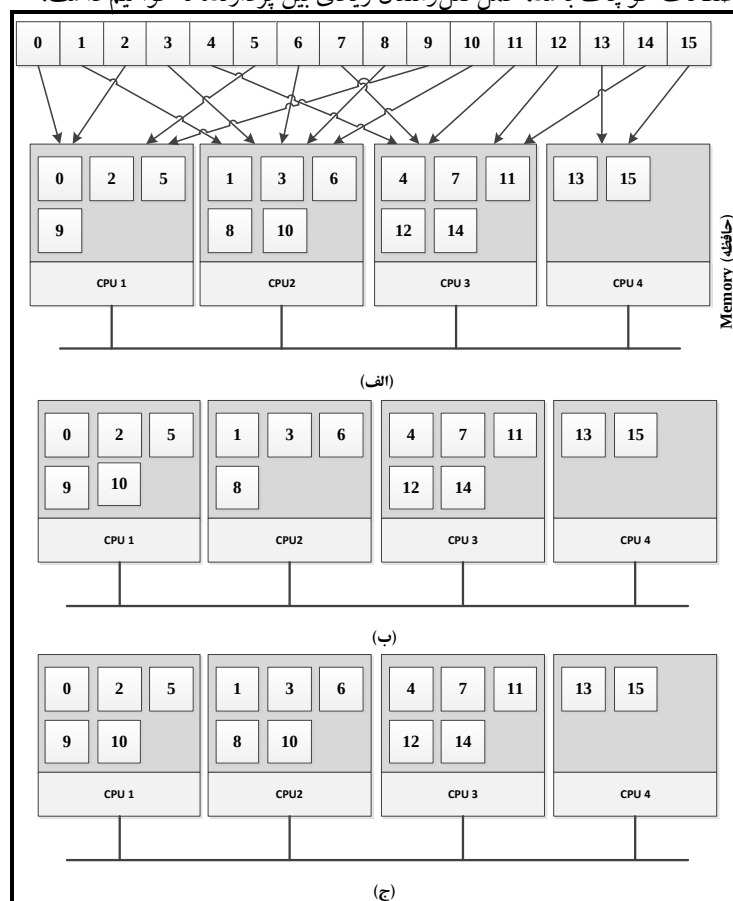
^۶. Page Replication

^۷.Coherency

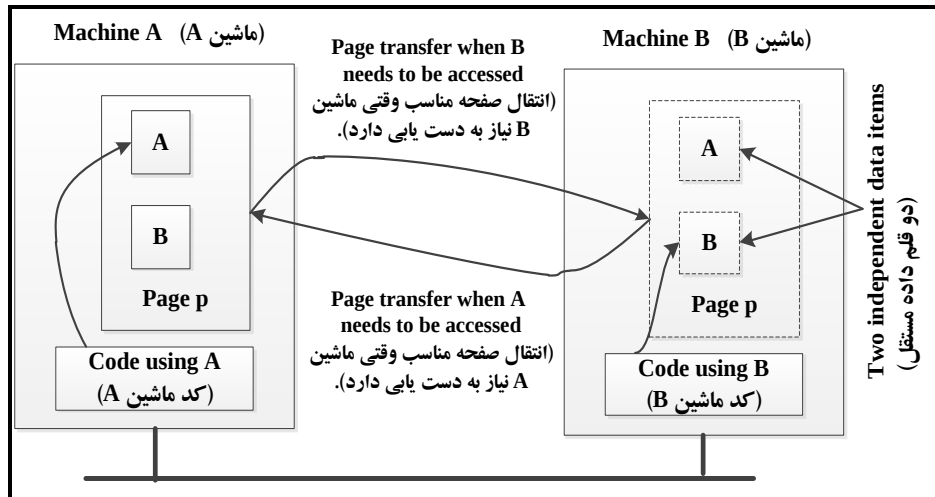
^۸.Consistency

صفحه‌ای و ۴ پردازنده را می‌بینید (پردازنده‌ها می‌خواهند به صورت توزیع شده عمل کنند و قسمتی از حافظه اشتراکی را دست‌یابی داشته باشند). در سیستم DSM، ۱۶ صفحه اشتراکی بین چهار پردازنده تقسیم شده است (ممکن است تعداد صفحات پردازنده‌ها مساوی نباشد). همان‌طور که در شکل (الف) می‌بینید، پردازنده ۱ چهار صفحه، پردازنده ۲، پنج صفحه، پردازنده ۳، پنج صفحه و پردازنده ۴، دو صفحه دارد. حال فرض کنید پردازنده ۱ نیاز به صفحه ۱۰ را داشته باشد. چون این صفحه به عنوان صفحه محلی پردازنده ۱ نیست (صفحه محلی پردازنده ۲ است)، نقض صفحه رخ می‌دهد. حال باید یکی از عملیات انتقال صفحه یا تکرار صفحه انجام شود. اگر انتقال صفحه انجام شود (شکل ب)، صفحه شماره ۱۰ از پردازنده ۲ به پردازنده ۱ منتقل می‌شود. اما اگر تکرار صفحه استفاده شود، در پردازنده ۲ یک صفحه محلی با همان محتوی صفحه شماره ۱۰ در پردازنده ۱ ایجاد می‌شود (شکل ج). این صفحه قابل تغییر نیست. چون اگر صفحات قابل نوشتن باشند، مشکل سازگاری را خواهیم داشت. به عنوان مثال، ممکن است محتوی صفحات هم شماره یکسان نباشد. نکته مهم دیگر در طراحی سیستم‌های DSM کارآمد، اندازه صفحات است. زیرا:

🌈 اگر اندازه صفحات کوچک باشد، عمل نقل و انتقال زیادی بین پردازنده‌ها خواهیم داشت.



شکل ۹-۱ سیستم حافظه اشتراکی توزیع شده (DSM).



شکل ۱۰-۱ نمایش عمل False sharing یک صفحه بین دو فرآیند مستقل.

اگر اندازه صفحه بزرگ باشد، مشکل اشتراک اشتباه^۱ را خواهیم داشت. منظور از اشتراک اشتباه، وجود داده متعلق به دو پردازنده را در یک صفحه گویند (شکل ۱۰-۱). این موضوع باعث می‌شود تا آن صفحه مکرراً بین دو پردازنده که به آن نیاز دارند، ردوبدل شود. پس اگر اندازه صفحه بزرگ انتخاب گردد، نرخ انتقال بسیار زیادی در شبکه برای انتقال صفحه خواهیم داشت.

۹-۱. مقایسه انواع سیستم‌ها

مقایسه انواع سیستم‌ها را در جدول ۱-۱ می‌بینید.

۹-۱-۱. انواع سیستم‌عامل‌های توزیع‌شده

سیستم‌عامل‌های توزیع‌شده انواع مختلف دارند که عبارت‌اند از:

۱. سیستم‌عامل‌های توزیع‌شده (DOS)^۲ ۲. سیستم‌عامل شبکه (NOS)^۳

۳. سیستم‌عامل مبتنی بر میان‌افزار

جدول ۱-۱ مقایسه انواع سیستم‌ها.		
سیستم	توصیف	اهداف اصلی
DOS	سیستم‌عامل با ارتباطات قوی ^۴ برای چندپردازنده‌ای و چند کامپیوترهای همگن	مخفی و مدیریت کردن منابع سخت‌افزاری
NOS	سیستم‌عامل با ارتباط ضعیف ^۵ برای چند کامپیوتر ناهمگن (شبکه‌های LAN و WAN)	فراهم کردن سرویس‌های محلی برای سرویس‌گیرنده‌های راه دور
میان‌افزار	لایه‌ای بالای سیستم‌عامل شبکه (NOS) اضافه‌شده تا سرویس‌های با اهداف عمومی را پیاده‌سازی کند.	فراهم کردن شفافیت توزیع

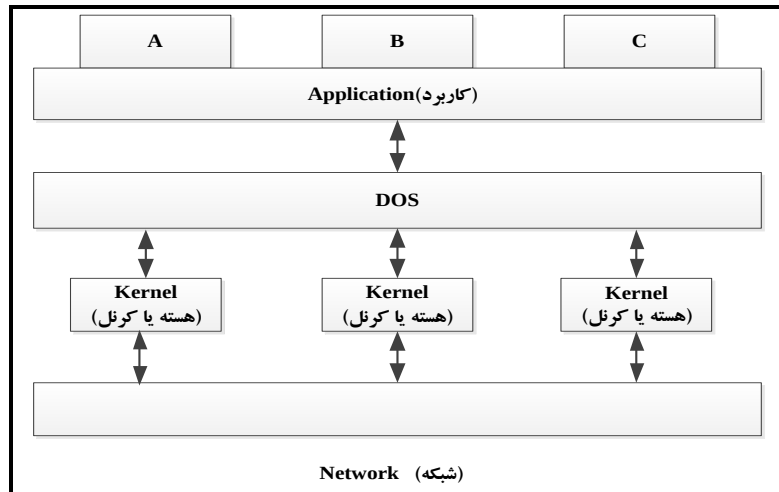
^۱.False sharing

^۲.Distributed Operating Systems

^۳.Network Operating Systems

^۴.Tightly- coupled

^۵.Loosely- coupled



شکل ۱۱-۱ سیستم عامل توزیع شده DOS و نقش DOS

سیستم عامل DOS

همان طور که می دانید سیستم عامل یکی از نرم افزارهای بسیار مهم است. به همین دلیل در این بخش به طور مختصر به سیستم عامل می پردازیم. سیستم عامل می تواند بر روی یک پردازنده نصب شود. در این صورت سیستم عامل می تواند دارای ساختار یکپارچه باشد (مانند شکل ۱۱-۱) یا با ساختار میکرو کرنل باشد. در سیستم عامل با ساختار یکپارچه، مدیریت حافظه، فرآیند و مدیریت فایل همه توسط کرنل (هسته) انجام می شود. بنابراین، دارای مشکلات زیر (همان مشکلات کرنل) است:

۱. قابلیت انعطاف^۱ آن پایین است. ۲. قابلیت استفاده مجدد^۲ آن نیز پایین است.
۳. توسعه پذیری^۳ آن پایین است. زیرا، اگر قسمتی از کرنل تغییر کند، کل کرنل باید تغییر یابد.
- برای رفع برخی از این مشکلات ساختار میکرو کرنل ایجاد شده اند (شکل ۱۲-۱). میکرو کرنل شامل دستوراتی است که باید در مد کرنل اجرا شود. به عبارتی می توان گفت شامل کدی برای تنظیم کردن ثباتها، سوئیچ کردن بین فرآیندها و مدیریت MMU^۴ و به دست آوردن وقفه های سخت افزاری شامل کدی جهت پاس کردن فراخوانی های سیستمی به ماژول مناسب سیستم عامل می باشد. مزایای میکرو کرنل عبارتند از:
۱. قابلیت انعطاف را افزایش داده است. چون همان طور که از شکل ۱۲-۱ می بینید، اگر بخواهید یکی از ماژولها را عوض کنید، نیاز به ترجمه (کامپایل) یا نصب مجدد سیستم نیست. به عنوان مثال، می توان ماژول کاربرده^۵ را در یک ماشین قرار داد و ماژول فایل را در ماشین دیگری قرار داد.
۲. میکرو کرنل را به راحتی می توان برای سیستم های توزیع شده استفاده نمود.

چند پردازنده ای سیستم عاملش همان سیستم عامل تک پردازنده است، با این تفاوت که مدیریت منبع دیگری به نام پردازنده^۶ نیز وجود دارد که جهت هم زمانی می توان از ابزارهای هم زمانی مانند سمافور و مانیتور استفاده کرد که در درس سیستم عامل دیده اید.

^۱.Flexibility

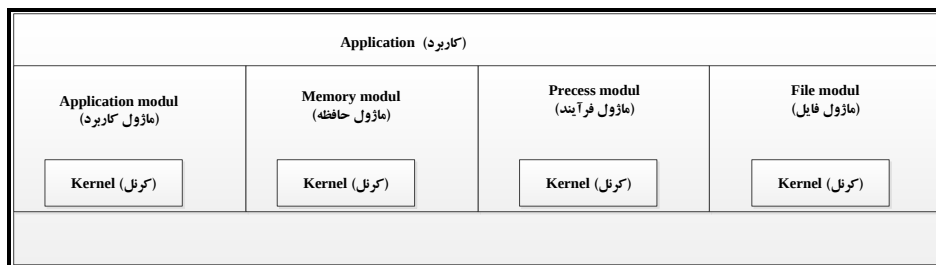
^۲.Reusability

^۳.Scalability

^۴.Memory Management Unit

^۵.Application

^۶.CPU



شکل ۱۲-۱ ساختار میکرو کرنل.

در سیستم‌های چند کامپیوتری سیستم‌عامل توزیع‌شده است و باید همگن باشد (سیستم‌های توزیع‌شده بر روی سیستم‌های همگن قرار می‌گیرند و با سیستم‌عامل‌های معمولی فرق دارد. نمونه‌ای از سیستم‌عامل توزیع‌شده مثل DOS را در شکل ۱۲-۱ می‌بینید. همان‌طور که در این شکل مشاهده می‌شود، هر ماشین کرنل خاص خودش را دارد که روی همه این‌ها DOS قرار می‌گیرد. نقش اصلی DOS، پیاده‌سازی نرم‌افزاری حافظه مشترک است و وظایفی از قبیل انتساب وظایف فرآیندها، پوشش خطای سخت‌افزاری پنهان‌سازی ذخیره‌سازی و ارتباطات سراسری بین فرآیندها را به عهده دارد. اما، کرنل، همان وظایف کرنل معمول از قبیل مدیریت حافظه محلی، مدیریت CPU محلی، مدیریت دیسک محلی و غیره را به عهده دارد.

سیستم‌عامل شبکه

برخلاف سیستم‌عامل‌های توزیع‌شده، سیستم‌عامل‌های شبکه می‌توانند بر روی سخت‌افزارهای ناهمگن (از انواع مختلف) اجرا شوند و عمل مخفی‌سازی در این سیستم‌ها وجود ندارد. به‌عنوان مثال، در هنگام کپی فایل در شبکه، مشخص است از کدام کامپیوتر کپی می‌شود، این سیستم‌ها از مجموعه‌ای از سیستم‌های تک‌پردازنده‌ای ایجادشده‌اند که هر کدام سیستم‌عامل خاص خودش را دارند (شکل ۱۳-۱). همان‌طور که در این شکل می‌بینید، ماشین‌ها و سیستم‌عاملشان ممکن است متفاوت باشند. اما، تمامی این ماشین‌ها از طریق شبکه با هم ارتباط دارند و سیستم‌عامل شبکه^۱ (NOS) امکاناتی را فراهم می‌کند تا به کاربران اجازه دهد که از سرویس موجود در یک ماشین خاص استفاده کنند.

تفاوت دیگر در این شبکه این است که سرویس‌گیرنده‌ها می‌توانند از طریق آدرس‌های مختلف (مسیرهای متفاوت) به سرویس‌دهنده‌ها دسترسی داشته باشند. شکل ۱۳-۱ این موضوع را نشان می‌دهد. همان‌طور که در این شکل می‌بینید، یکی از سرویس‌دهنده‌ها دایرکتوری به نام Games را در خود دارد و سرویس‌دهنده دیگر دایرکتوری به نام work، هر کدام از این دایرکتوری‌ها شامل چندین فایل می‌باشند. این دو سرویس‌گیرنده به هر دو سرویس‌دهنده نصب هستند (اما با مسیرهای متفاوت). درواقع، سرویس‌گیرنده ۱ به دایرکتوری اصلی آن‌ها دسترسی دارد و می‌تواند به شکل Games و Work به سرویس‌دهنده‌ها دسترسی داشته باشد. سرویس‌گیرنده ۲ نیز به دایرکتوری اصلی سرویس‌دهنده ۲ دسترسی دارد (مانند سرویس‌دهنده ۱). اما با این دید که Games شاید به‌عنوان کار خصوصی^۲ باشد، سرویس‌گیرنده ۲ یک دایرکتوری به نام Private/ایجاد می‌کند و از طریق آن به سرویس‌دهنده‌ی که دایرکتوری Games را دارد، دسترسی پیدا می‌کند.

^۱.Network Operating System

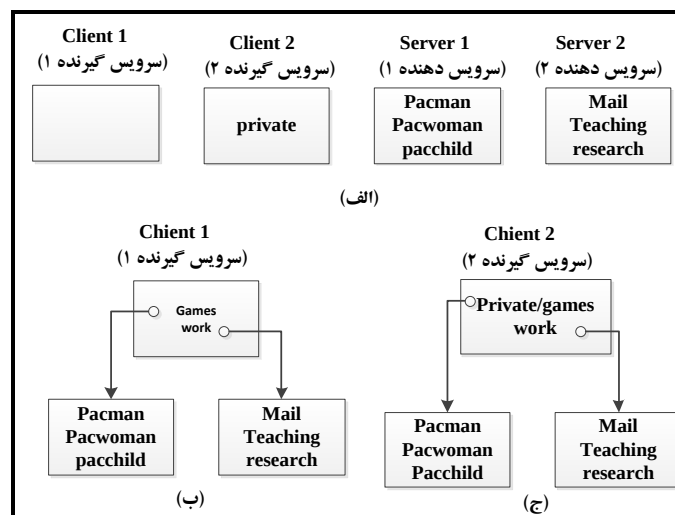
^۲.Private

در نتیجه، اگر سرویس گیرنده ۲ بخواهد به Packwoman دسترسی داشته باشد، مسیر به صورت Private/Games/Packwoman خواهد بود. ولی سرویس گیرنده ۱ با مسیر Games/Packwoman به همین دایرکتوری Games دسترسی دارد.

در سیستم عامل شبکه درجه شفافیت پایین است، در سیستم عامل شبکه ماشین ها می توانند ناهمگن باشند. در سیستم عامل شبکه N کپی از سیستم عامل داریم. در سیستم عامل شبکه ارتباطات مبتنی فایل است. در سیستم عامل شبکه مدیریت منابع برحسب گروه ها است. سیستم عامل شبکه توسعه پذیری دارد. سیستم عامل شبکه، باز بودن^۱ را پشتیبانی می کند.

سیستم عامل های میان افزار

همان طور که بیان گردید، هیچ کدام از سیستم عامل های شبکه، سیستم عامل های توزیع شده، سیستم عامل توزیع شده واقعی نیستند. چون سیستم عامل توزیع شده نمی تواند مجموعه ای از کامپیوترهای مستقل را پشتیبانی کند و سیستم عامل شبکه یک دید واحدی را از سیستم به کاربر ارائه نمی دهد. اکنون این سؤال مطرح است آیا می توان سیستم عامل توزیع شده را ارائه کرد که قادر به اداره کامپیوترهای مستقل از هم باشد و دید واحدی از سیستم را ارائه نماید. به عبارت دیگر، سیستم عامل توزیع شده ای که قابلیت توسعه پذیری و باز بودن سیستم عامل شبکه را داشته باشد و قابلیت مخفی سازی و ارتباط آسان سیستم عامل توزیع شده را نیز دارا باشد. برای این منظور، یک لایه نرم افزاری اضافی در سیستم عامل های شبکه استفاده می کنیم، به طوری که این لایه باید ناهمگنی های سخت افزاری را مخفی کرده و برای ایجاد شفافیت (پنهان سازی) توسعه داده شود. این لایه، میان افزار^۲ نام دارد که در شکل ۱-۱ جایگاه میان افزار را دیده اید. ویژگی های سیستم عامل های مبتنی میان افزار را در زیر می بینید:



شکل ۱-۱۳ روش های مختلفی که سرویس گیرنده به سرویس دهنده های در مکان های متفاوت نصب می شوند.

^۱.Openness ^۲.Middle ware

جدول ۲-۱ مقایسه سیستم‌های مختلف.				
قلم	سیستم عامل توزیع شده		سیستم عامل شبکه	سیستم عامل مبتنی بر میان افزار
	چند پردازنده ای	چند کامپیوتری		
درجه شفافیت	خیلی بالا	بالا	پایین	بالا
آیا سیستم عامل همه گره‌ها همسان است.	بله	بله	نه	نه
تعداد کپی سیستم عامل	۱	N	N	N
مبنای برقراری ارتباطات	حافظه مشترک	پیام‌ها	فایل‌ها	مدل مشخص شده
مدیریت منابع	عمومی و متمرکز	عمومی و توزیع شده	بر اساس گره	بر اساس گره
توسعه پذیری	ندارد	سخت	بله	خیلی زیاد
باز بودن سیستم	بسته	بسته	باز	باز

سیستم توزیع شده مبتنی بر میان افزار درجه شفافیت بالایی دارد.

در سیستم توزیع شده مبتنی بر میان افزار گره‌ها می‌توانند همگن نباشند.

در سیستم توزیع شده مبتنی بر میان افزار تعداد کپی سیستم عامل N است.

در سیستم عامل مبتنی بر میان افزار پایه ارتباطات مدل مشخص شده است.

مدیریت منابع در سیستم عامل توزیع شده مبتنی بر میان افزار در اختیار گره‌ها است.

سیستم عامل توزیع شده مبتنی بر میان افزار توسعه پذیری بالایی دارد.

سیستم عامل توزیع شده مبتنی بر میان افزار باز بودن سیستم را پشتیبانی می‌کند.

۲-۹-۱. مقایسه سیستم‌ها با یکدیگر

همان‌طور که بیان گردید، سیستم‌های مختلف مزایا و معایبی دارند. مزایا و معایب سیستم‌های مختلف را در جدول ۲-۱ می‌بینید.

۳-۹-۱. تله‌ها (دام‌ها) در سیستم‌های توزیع شده

تله‌ها (دام‌ها)^۱، فرض‌ها نادرستی هستند که در سیستم‌های توزیع شده وجود دارند. برخی از این تله‌ها عبارت‌اند از:

۱. شبکه قابل اعتماد است. ۲. شبکه امن است. ۳. شبکه همگن^۲ است.
۴. توپولوژی تغییر نمی‌کند. ۵. تأخیر^۳ صفر است. ۶. پهنای باند نامحدود^۴ است. ۷. هزینه انتقال صفر است. ۸. فقط یک سرپرست^۵ وجود دارد.

۱۰-۱. انواع سیستم‌های توزیع شده

سیستم‌های توزیع شده به سه دسته تقسیم می‌گردند که عبارت‌اند از:

۱. سیستم‌های محاسباتی توزیع شده (Distributed Computing Systems)

^۱.pitfills

^۲.Homogenous

^۳.Latency

^۴.Infinite

^۵.Administrator

۲. سیستم‌های اطلاعاتی توزیع شده (Distributed Information Systems)

۳. سیستم‌های فراگیر توزیع شده (Distributed Embedded Systems)

۱-۱۰-۱. سیستم‌های محاسباتی توزیع شده

سیستم‌های محاسباتی توزیع شده برای کارهای محاسباتی با کارایی بالا به کار می‌روند. این سیستم‌ها به دو گروه زیر تقسیم می‌گردند:

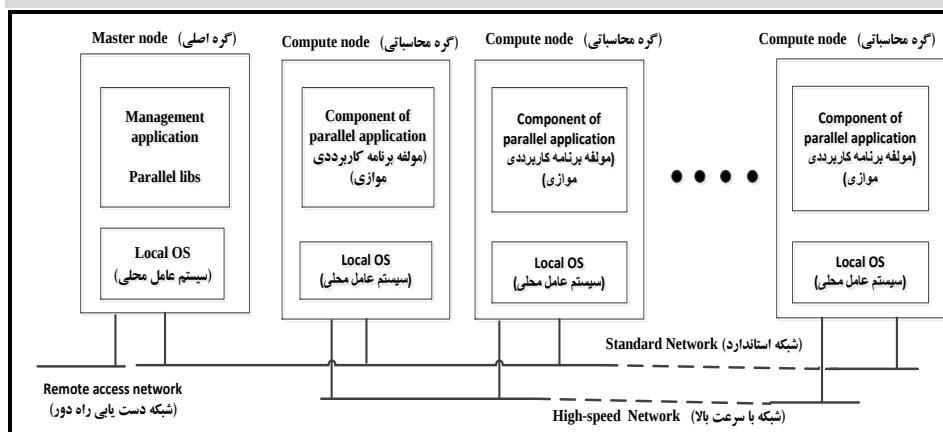
۱. **سیستم‌های محاسباتی خوشه‌ای**^۱، از طریق سخت‌افزارهایی که شامل مجموعه‌ای از کامپیوترهای شخصی یا ایستگاه‌های کاری^۲ همگن و سیستم‌عامل‌های یکسان هستند که از طریق شبکه محلی سریع به هم متصل هستند. نمونه‌ای از سیستم‌های محاسباتی خوشه‌ای در شکل ۱۴-۱ آمده است. سیستم‌های خوشه‌ای محاسباتی دارای ویژگی‌های زیر هستند:

۱. هر خوشه شامل مجموعه‌ای از گره‌های محاسباتی است. گره‌های محاسباتی همگن هستند و سیستم‌عاملشان یکسان است (گره کامپیوتر در شکل، گره محاسباتی هستند). گره محاسباتی فقط سیستم‌عامل استاندارد را اجرا می‌کند.

۲. گره‌ها محاسباتی و گره اصلی از طریق شبکه محلی پرسرعت به هم متصل هستند. گره‌ها شبکه محلی پرسرعت به هم متصل هستند. گره‌ها با گره اصلی در تماس هستند.

۳. هر خوشه شامل یک گره اصلی^۳ است. وظایف گره اصلی عبارت‌اند از: ۱. تخصیص گره‌ها به برنامه موازی خاص ۲. تشکیل صف کارها^۴ ۳. ارائه واسطی^۵ برای کاربران. پس گره اصلی، میان‌افزار مورد نیاز برای اجرای برنامه و مدیریت خوشه است.

۴. در این نوع سیستم‌ها، فاکتور کارایی / قیمت^۶ بهبود می‌یابد (یعنی، این فاکتور کاهش می‌یابد). بنابراین به کمک این ابزار می‌توان ابر کامپیوتر را ساخت.



شکل ۱۴ - ۱ نمونه‌ای سیستم محاسباتی خوشه‌ای Beowulf

^۱.Cluster Computing Systems ^۲.Workstation ^۳.Master Node ^۴.Jobs ^۵.Interface
^۶.Price/ Performance

۵. هدف محاسبات خوشه‌ای، محاسبات موازی است. یعنی، یک گره وظیفه را به اجزا کوچک‌تری تقسیم می‌کند تا هر گره بتواند یک وظیفه را انجام دهد. در محاسبات خوشه‌ای، چند وظیفه‌گی^۱ نداریم.

۶. هزینه سیستم‌های محاسباتی خوشه‌ای بالا است.

🌐 **سیستم‌های محاسباتی شبکه‌ای (گرید)^۲**، مجموعه‌ای از کامپیوترهای ناهمگن (از لحاظ سخت‌افزاری، سیستم‌عامل، شبکه و امنیت) هستند که مهم‌ترین هدف آن‌ها قرار دادن آسان منابع در اختیار برنامه‌های کاربردی است. برخی از ویژگی‌های سیستم‌های محاسباتی شبکه‌ای (گرید) عبارت‌اند از:

۱. گره‌ها ناهمگن هستند (یعنی، ممکن است سخت‌افزار، سیستم‌عامل شبکه و حتی امنیت آن‌ها متفاوت باشد).

۲. منابع در سازمان‌های مختلف پراکنده‌اند و تحت سرپرستی‌های مختلف هستند.

۳. گروهی از برنامه‌های و کاربران وجود دارند که از منابع مختلف برای انجام وظیفه استفاده می‌کنند. این گروه کاربران، سازمان مجازی^۳ نام دارند.

۴. توسعه‌پذیری در سیستم‌های محاسباتی گرید بهتر از سیستم‌های محاسباتی خوشه‌ای است. اما، در کارهای محاسباتی بهتر است سیستم‌های محاسباتی خوشه‌ای استفاده شود. نمونه‌ای از سیستم‌های محاسباتی گرید در شکل ۱۵-۱ آمده است. کاربردها^۴، همان سازمان مجازی هستند و از گرید برای اجرا استفاده می‌کنند که به منابعی نیاز دارند. همان‌طور که در این شکل می‌بینید، این سیستم از چهار لایه تشکیل شده است که عبارت‌اند از:

🌐 **لایه تجمیع^۵**، اداره مجموعه‌ای از منابع را به عهده دارد. شامل سرویس‌هایی جهت کشف (یعنی، تقسیم در مورد این که کدام سرویس کارایی بهتر و هزینه کم‌تری دارد)، تخصیص و زمان‌بندی وظایف در چندین منبع، اداره کردن تکثیر داده^۶ (برای افزایش کارایی^۷ و تحمل خرابی^۸) و غیره است.

🌐 **لایه اتصال^۹**، شامل پروتکل‌های ارتباطی است که برای پشتیبانی از تراکنش‌های گرید لازم می‌باشد را انجام می‌دهد. به عنوان مثال، برای انتقال داده بین منابع و پروتکل‌های دیگر مانند پروتکل دسترسی راه دور^{۱۰} یا پروتکل امنیتی تأیید هویت کاربران یا برنامه‌های که از طرف آن‌ها اجرا می‌شود، به کار می‌رود. این لایه بهره‌وری از چندین منبع را ممکن می‌سازد.

🌐 **لایه منبع^{۱۱}**، برای مدیریت منابع به کار می‌رود. این لایه از توابعی که در لایه اتصال برای ارتباط مشخص شده است، پیاده‌سازی می‌کند و از آن‌ها استفاده می‌نماید. واسطه لایه زیرساخت^{۱۲} را فراخوانی می‌کند.

🌐 **لایه زیرساخت^{۱۳}**، مسئول تنها یک منبع است و حالت یک منبع را پرس‌وجو^{۱۳} می‌کند. این لایه نزدیک‌ترین لایه به لایه منبع است.

لایه‌های زیرساخت، منبع اتصال و تجمیع با یکدیگر میان‌افزار گرید را تشکیل می‌دهند. میان‌افزار گرید، وظیفه دسترسی و مدیریت منابع را انجام می‌دهد. یکی از مهم‌ترین ملاحظات از لحاظ میان‌افزار این است که در

¹.Multi Tasking ².Grid Computing System ³.Virtual Organization ⁴.Applications
⁵. Collective Layer ⁶.Data replication ⁷.Performance ⁸.Falut ⁹.Fiberlayetelorance
¹⁰. Connectivity layer ¹¹.Remote Acce ¹².Resource layess-protocol ¹³.Query

محاسبات گرید همیشه ایده سایت یا یک واحد به چشم می خورد. این جهت گیری به تدریج موجب شکل گیری معماری سرویس گرا^۱ (SOA) شده است. معماری سرویس گرا، امکان دسترسی به لایه های مختلف را از طریق مجموعه ای از سرویس ها وب فراهم می کند. معماری سرویس گرا نیز موجب ایجاد معماری دیگری به نام معماری سرویس گرا گرید باز (OGSA) شده است. این معماری از لایه ها و قطعات مختلفی تشکیل شده است که موجب پیچیدگی آن گردید.

۲-۱۰-۱. سیستم های اطلاعاتی توزیع شده

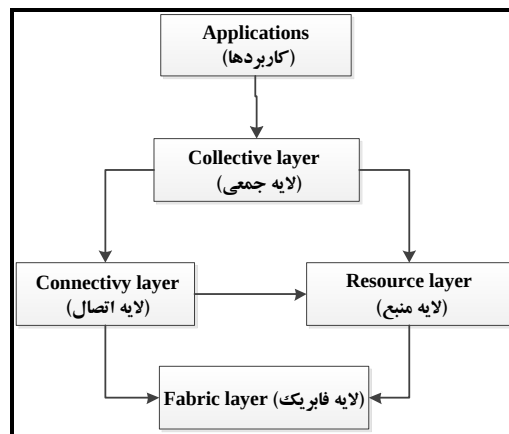
سیستم های اطلاعاتی توزیع شده برای شرکت ها یا سازمان های بزرگ که با برنامه های کاربردی شبکه ای سروکار دارند، استفاده می شود. سیستم های اطلاعاتی توزیع شده دو نوع اند که عبارت اند از:

۱. سیستم های پردازش تراکنش^۲، تراکنش فعالیتی است که دارای خاصیت همه یا هیچ است. فعالیت می تواند یک، چند و یا بخشی از فرآیند^۳ باشد. نمونه ای از تراکنش می توان انتقال پول از طریق عابر بانک از یک حساب به حساب دیگر را نام برد. این فعالیت به دو عمل برداشت پول از حساب اول و واریز پول به حساب دوم تجزیه می شود. در هنگام انتقال پول، چنانچه برداشت از حساب اول انجام شود، اما، واریز به حساب دوم (به هر دلیلی) انجام نگیرد، تأثیر برداشت پول از حساب او باید لغو شود (یعنی، تراکنش یا همه دستور آن باید انجام شوند، یا هیچ یک از دستورات انجام نشوند). تراکنش دارای چهار خاصیت است. این خواص ACID نام دارند که عبارت اند از:

۱. خاصیت یکپارچگی یا تجزیه ناپذیری^۴، یا همه یا هیچ است. یعنی، تراکنش قابل تجزیه نیست (مانند مثال انتقال پول از یک حساب به حساب دیگر).

۲. خاصیت سازگاری^۵، یعنی خاصیت یکپارچگی تراکنش، تغییرناپذیری تراکنش را نقض نمی کند.

۳. خاصیت انزوا^۶، تراکنش های هم زمان اثر مخرب بر روی یکدیگر نداشته باشند (با یکدیگر تداخل نداشته باشند).



شکل ۱-۱۵ نمونه ای از معماری لایه ای سیستم های محاسباتی گرید.

^۱.Service oriented Architecture ^۲.Distributed Trasaction Systems ^۳.Process ^۴.Atomic
^۵.Consistency ^۶.Isolation

جدول ۳-۱ دستورالعمل‌های یک تراکنش.	
دستورالعمل	هدف
BEGIN-TRANSACTION	ابتدای (شروع) تراکنش را تعیین می‌کند.
END-TRANSACTION	پایان تراکنش را مشخص کرده و آن را ثبت می‌کند.
ABORT-TRANSACTION	تراکنش را ساقط کرده و اثر تراکنش انجام‌شده را از بین می‌برد.
READ	داده‌هایی را از فایل، جداول و غیره می‌خواند.
WRITE	داده‌ها را در جدول، فایل و غیره می‌نویسد.

۴. خاصیت پایایی (پایداری)^۱، وقتی تراکنش تثبیت^۲ گردید، باید اثر آن دائمی بماند و از بین نرود.

دستورالعمل‌هایی که یک تراکنش باید پشتیبانی کند، در جدول ۳-۱ آمده است.

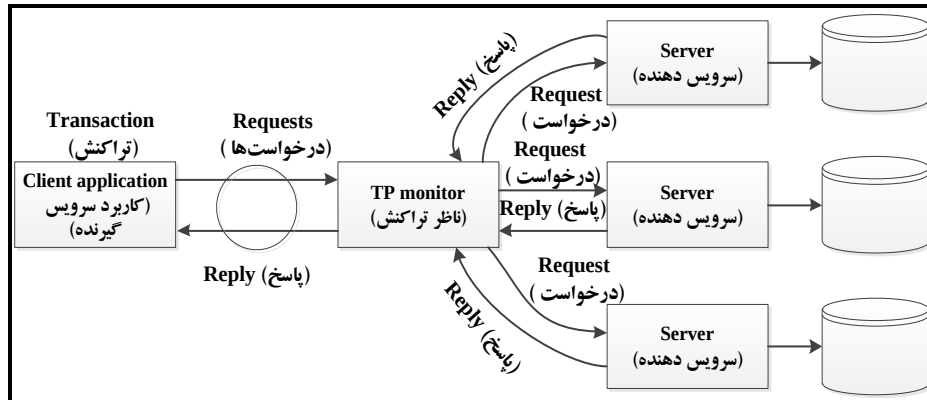
جامعیت کاربرد شرکت، همان‌طور که در پردازش تراکنش^۳ ملاحظه کردید، پردازش تراکنش منحصرراً از مدل سرویس‌دهنده / سرویس‌گیرنده^۴ استفاده می‌کرد. اما تراکنش‌های وجود دارند که به بخش‌های مختلف تقسیم می‌گردند. این تراکنش‌ها، **تراکنش‌های تودرتو**^۵ نام دارند (در ادامه این تراکنش‌ها را می‌بینید). تراکنش‌های تودرتو به متدهایی نیاز دارند تا کاربردها با هم دیگر ارتباط برقرار نمایند (یعنی، ارتباطات فقط به‌صورت سرویس‌دهنده / سرویس‌گیرنده نیست). روش‌های مختلفی برای ارتباط بین کاربردها وجود دارد که در ادامه مشاهده می‌کنید.

تراکنش‌های تودرتو، تراکنش‌های هستند که برخی از محدودیت‌های تراکنش سخت^۶ را حل می‌کنند. مهم‌ترین مشکل تراکنش سخت این است که اجازه نمی‌دهد، بخشی از دستورات تثبیت یا ساقط^۷ شود. یک تراکنش تودرتو، از تعدادی زیر تراکنش (تراکنش فرعی) تشکیل شده است (مانند شکل ۱۶-۱). یکی دیگر از مزایای تراکنش‌های تودرتو، امکان اجرای تراکنش‌ها به‌صورت توزیع‌شده روی چند کامپیوتر را فراهم می‌کنند. تراکنش سطح بالاتر، به چند تراکنش مختلف تقسیم می‌شود تا کارایی را بالا برده، برنامه‌نویسی را آسان‌تر کرده و تراکنش‌های فرزند را بتوان روی ماشین‌های مختلف به‌صورت موازی اجرا کرد، در این نوع تراکنش‌ها (تراکنش‌های تودرتو)، همان‌طور که در شکل می‌بینید، همه درخواست‌ها از طرف سرویس‌گیرنده-ها به یک موجودیت به نام **ناظر پردازش تراکنش (TPM)**^۸ اداره می‌شوند. سپس ناظر پردازش تراکنش، تراکنش را به چند تراکنش فرزند تقسیم می‌کند و به سرویس‌دهنده می‌فرستد. سپس، اگر همه تراکنش‌های فرزند، تثبیت شده باشند، تراکنش اصلی تثبیت می‌شود. اما، اگر یکی از تراکنش‌های فرزند ساقط شود، تمامی تراکنش‌های فرزند تثبیت شده باید عقب‌گرد کنند. **عقب‌گرد**^۹، یعنی این که اثر تراکنش‌های انجام‌شده، از بین برود.

در تراکنش‌های تودرتو از دیدگاه میان‌افزاری برنامه کاربردی وجود دارد.

انواع مختلف میان‌افزار وجود دارد که عبارت‌اند از:

^۱. Durable ^۲.Commit ^۳.Transaction Processing ^۴.Client/server ^۵.Nested Transactions
^۶. Flat transaction ^۷.Abort ^۸.Sub Transaction ^۹.Rollback



شکل ۱۶- ۱ نقش یک ناظر پردازش تراکنش در تراکنش‌های توزیع شده.

۱. فراخوانی رویه راه دور (RPC)^۱، به برنامه اجازه می‌دهد تا رویه‌های موجود در کامپیوتر دیگر فراخوانی گردد. این روش از طریق ارسال پیام و برگشت نتیجه فراخوانی پیاده‌سازی می‌شود.
۲. احضار متد راه دور (RMI)^۲، مانند RPC می‌باشد. با این تفاوت که برای سیستم‌های شی گرا استفاده می‌شود.

۳. میان‌افزار پیام گرا (MOM)^۳، در این روش کاربردها از طریق پیام با هم دیگر ارتباط برقرار می‌کنند. شرح کامل روش‌های مختلف ارتباطات RPC، RMI و MOM را در فصل چهارم می‌بینید.

۳- ۱۰- ۱. سیستم‌های فراگیر توزیع شده

همان‌طور که در سیستم‌های توزیع شده محاسباتی و اطلاعاتی ملاحظه کردید، گره‌ها پایدار بودند و از طریق شبکه‌های مناسب به یکدیگر متصل بودند (یعنی، محل گره‌ها تقریباً ثابت بود، کیفیت اتصال و خرابی در سیستم معمولاً نرمال می‌باشد). اما در دنیای امروزی سیستم‌های متحرک و فراگیر (تعبیه شده)^۴ بسیار زیادی استفاده می‌شوند. این سیستم‌ها معمولاً **اتصالات بی سیم** دارند. متحرک هستند، کوچک هستند، محاسبات کمی انجام می‌دهند و با باتری کار می‌کنند. پس بخش عظیم پیکربندی و ایجاد تغییرات وضعیتی آن‌ها باید به صورت خودکار انجام شود.

این سیستم‌ها دارای سه نیاز هستند:

۱. پذیرفتن **تغییرات موقعیتی**، یعنی باید بدانند که محیط اطرافش به طور دائم در حال تغییرات و باید این تغییرات را تشخیص داده و عمل مناسب در برابر آن انجام دهد (مثل عدم دسترسی به شبکه در تلفن‌های همراه).
۲. **تقویت ترکیب‌های ویژه**، یعنی دستگاه‌های متعدد توسط کاربران مختلف در این سیستم استفاده می‌شوند. برنامه‌های کاربردی که در این دستگاه‌ها اجرا می‌شوند، باید به راحتی توسط کاربران یا به صورت خودکار پیکربندی^۵ شوند (مانند پیکربندی GPRS در دستگاه‌های مختلف تلفن‌های همراه).

^۱.Remote Procedure Call ^۲.Remote Method Invocation ^۳.Message oriented middleware
^۴.pervasive (Embedded) ^۵.Self Configuration

۳. تشخیص اشتراک به صورت پیش فرض، یعنی در اتصال دستگاه به سیستم که هدف اصلی آن خواندن و نوشتن داده‌ها است، فضای نوشتن و خواندن داده‌ها به صورت اشتراک مورد استفاده قرار می‌گیرد. با توجه به این که اتصال متغیر است و متناوب در حال تغییر می‌باشد، این فضای اشتراکی در هر اتصال باید به درستی تشخیص داده شود. در سیستم‌های فراگیر توزیع شده، ایجاد شفافیت سخت‌تر از بقیه سیستم‌ها است و با توجه به سیار بودن دستگاه‌ها، گاهی بهتر است که برخی از شفافیت‌ها لحاظ نشود. چون موجب کاهش کارایی سیستم می‌شود. نمونه‌های از سیستم‌های فراگیر توزیع شده عبارت‌اند از:

- سیستم‌های خانگی (Home Systems)
- سیستم سلامت الکترونیکی (Electronic Health Care Systems)
- شبکه‌های حسگر (Sensor Networks)

سیستم‌های خانگی

در این نوع سیستم‌ها یک شبکه خانگی وجود دارد که سیستم‌های خانگی نظیر یک یا چند کامپیوتر شخصی، PDA، سیستم‌های صوتی و تصویری (تلفن و تلویزیون)، سیستم حرارتی، سیستم نظارت و نظایر آن در آن به هم متصل هستند و صاحب سیستم می‌تواند آن‌ها را از بیرون خانه و از طریق اینترنت کنترل و مدیریت کند. این سیستم‌ها باید دو ویژگی زیر را داشته باشند:

۱. چون محیط دائم در حال تغییر است، سیستم‌ها باید خود پیکر بند باشند.
۲. سیستم‌ها باید خود را مرتب کنند. یعنی ورود و خروج از سیستم اداره شود. به عبارت دیگر، بسیاری از کارها انجام می‌شوند تا دستگاه‌ها به طور خودکار و از طریق استانداردهای نصب و اجرای جهانی (upnp)^۲ آدرس‌ها را به دست آورده و هم دیگر را پیدا می‌کنند.

سیستم‌های سلامت الکترونیکی

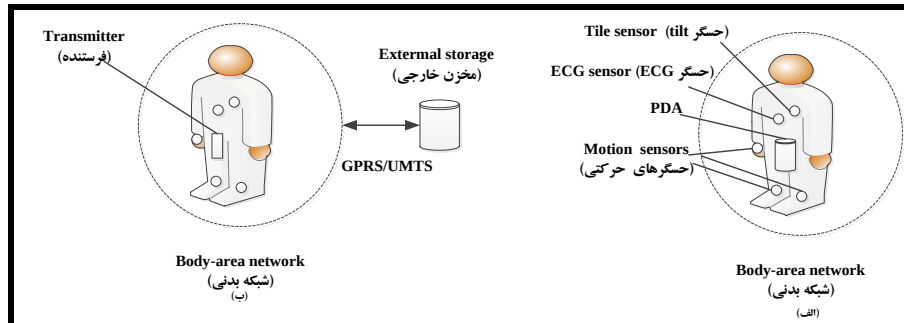
در این سیستم‌ها معمولاً یک سری حسگرهای مختلف در سطح بدن انسان توزیع می‌شوند تا سلامت بدن را کنترل و مانیتور کنند. این سیستم‌ها به طور خودکار سلامت افراد را نظارت می‌کنند و در صورت لزوم با پزشک تماس می‌گیرند. دو نوع این سیستم وجود دارد که عبارت‌اند از (شکل ۱۷-۱):

۱. هاب‌دار، در این نوع سیستم یک هاب وجود دارد که هر چند وقت به چند وقت اطلاعات را به کامپیوتر می‌دهد (هاب، می‌تواند مدیریت شبکه را نیز انجام دهد).
۲. بدون هاب، در سیستم فرستنده وجود دارد که گره‌ها اطلاعات را به فرستنده می‌فرستند. این شبکه، شبکه ناحیه بدن (BAN)^۳ نام دارد.

از دیدگاه سیستم توزیع شده با سؤالات زیر مواجه هستیم:

- داده کجا باید ذخیره شوند.
- چگونه جلوی از دست رفتن داده‌های حیاتی را بگیریم.
- چه زیرساختی برای تولید و پخش هشدارها لازم است؟
- با توجه به این که شبکه ضعیف است، سیستم چگونه مقاوم و مستحکم باشد؟

^۱.Self Managing ^۲.Universe of plug and play ^۳.body Area network



شکل ۱۷-۱ نظارت بر فرد در سیستم سلامت الکترونیکی فراگیر (الف) هاب دار (ب) اتصال بی سیم.

امنیت چگونه برقرار می شود؟

پزشک چگونه بازخورد^۱ را به شخص بدهد؟

۱۱-۱. شبکه های حسگر

شبکه حسگر معمولاً شامل صدها و هزاران گره نسبتاً کوچک است. هر یک از گره ها مجهز به دستگاه حسگر هستند که درجه حرارت و عبور هر چیزی را حس می کنند. گره ها معمولاً دارای باتری هستند. محدودیت منابع و قابلیت ارتباطی (از طریق شبکه های بی سیم) و پایین بودن توان مصرفی در آن ها موجب شده است که در هنگام طراحی توجه ویژه ای به کارایی شود. در این شبکه دو وضعیت وجود دارد (در شکل ۱۸-۱ می بینید).

۱. گره ها با هم ارتباط ندارند و پردازش نیز انجام نمی دهند. اطلاعات را به یک کامپیوتر مرکزی می فرستند (شکل (الف)).

۲. گره ها توان پردازش و ذخیره محدود را دارند. سپس اطلاعات را به کامپیوتر مرکزی می فرستند (شکل ۱۸-۱(ب)).

داده حس شده توسط حسگرها به سه روش ارسال می شود:

➤ داده حس شده همراه^۲ ارسال می شود.

➤ داده حس شده هرزمانی که نیاز باشد، ارسال می شود. یعنی مبتنی بر پرس و جو^۳ است.

➤ وقتی عمل خاصی رخ دهد، داده حس شده ارسال می شود. یعنی مبتنی بر رویداد^۴ است.

۱۲-۱. سؤالات چهار گزینه ای

۱. کدام مورد جزو مزیت های سیستم توزیع شده نیست به یک سیستم مرکزی است (فراگیر - ۸۵).

الف: فاکتور $\frac{\text{قیمت}}{\text{کارایی}}$ (Price/ Performance) بالا، سرعت بالا، توزیع ذاتی برنامه ها، قابلیت اطمینان و امکان رشد سیستم.

ب: فاکتور $\frac{\text{قیمت}}{\text{کارایی}}$ (Price/ Performance) بالا، سرعت بالا، توزیع ذاتی برنامه ها، قابلیت اطمینان بالا و توسعه پذیری.

^۱.Feedback ^۲.Continues ^۳.Query Based ^۴.Event Based

آشنایی با سیستم‌های توزیع‌شده ۳۷

ج: فاکتور $\frac{\text{قیمت}}{\text{کارایی}}$ (Price/ Performance) بسیار کم، سرعت بالا، توزیع ذاتی برنامه‌ها، قابلیت اطمینان بالا و امکان رشد سیستم.

د: فاکتور $\frac{\text{قیمت}}{\text{کارایی}}$ (Price/ Performance) بسیار کم، سرعت بالا، توزیع ذاتی برنامه‌ها، قابلیت اطمینان بالا و توسعه پذیری (Scalability).

۲. کدام یک از مدل‌های زیر جامع‌تر و در پیاده‌سازی سیستم‌عامل توزیع‌شده به کار رفته است؟ (فراگیر - ۸۵).
الف: SPMD ب: MIMD ج: MPMD د: MIMD

۳. به نظر شما مدل حافظه یک مالتی کامپیوتر و مدل حافظه یک مالتی پروسور به ترتیب کدام است؟ (فراگیر - ۸۵).
الف: UMA, UMA ب: UMA, NUMA ج: NUMA, NUMA د: می‌توانند هر یک از موارد ۲ و ۳ باشد

۴. کدام مورد تعریف شفافیت (Transparency) صحیح است؟ (فراگیر - ۸۵).
الف: شفافیت به معنای آن است که خیلی از کارها به صورت ناپیدا از دید کاربر انجام شود.
ب: شفافیت مفهومی است که اگر در سیستم پیاده شود، موجب آسانی برنامه‌نویسی برای کاربر می‌شود.
ج: شفافیت مفهومی است که این امکان را به وجود می‌آورد که یک سیستم توزیع‌شده از دید کاربر به عنوان یک کامپیوتر واحد عمل نماید.
د: هر سه

۵. در یک مدیریت فایل توزیع‌شده شفافیت مکان (Location Transparency) به چه معنا است؟ (فراگیر - ۸۵).
الف: کاربر از محل قرار گرفتن فایل مورد نظرش اطلاعی ندارد.
ب: عملیات بدون اطلاع کاربر بتوانند به موازات هم دیگر انجام شوند.
ج: نسخه‌های متعددی از فایل مورد نظر کاربر در ماشین‌های مختلف شبکه وجود داشته باشد.
د: فایل‌ها بدون آن که نامشان عوض شود، می‌توانند تحت صلاح دید مدیریت فایل از ماشینی به ماشین دیگر منتقل شوند.

۶. در کدام یک از انواع شفافیت (Transparency) در یک DOS، کاربرهای چندگانه می‌توانند منابع داده-ای را به صورت خودکار به اشتراک بگذارند؟ (فراگیر - ۸۷).
الف: شفافیت مهاجرت (Migration Transparency)
ب: شفافیت موازی (Parallelism Transparency)
ج: شفافیت مکان (Location Transparency)
د: شفافیت هم‌زمانی (هم‌روندی) (Concurrency Transparency)

۷. کامل‌ترین تعریف از یک سیستم توزیع یافته (Distributed System) کدام است؟ (فراگیر - ۸۸).
الف: مجموعه چند کامپیوتر که برای انجام هدف مشخص با هم در ارتباط هستند.
ب: مجموعه چند کامپیوتر که مستقل بوده، ولی از دید کاربر در حکم یک سیستم واحد هستند.
ج: مجموعه چند کامپیوتر همگن (Homogenous) است که می‌توانند چندین برنامه مختلف را اجرا نمایند.
د: مجموعه از چند کامپیوتر یا زیرسیستم است که در یک محدوده فیزیکی تعریف شده با هم به صورت مستقل از محیط بیرون در تقابل هستند.

۸. عبارت <<یک منبع می تواند توسط چندین کاربر مختلف در رقابت باشد>> معرف کدام از انواع شفافیت (Transparency) برای سیستم توزیع یافته است؟ (فراگیر - ۸۸).

الف: هم رندی (Concurrency) ب: تکرار (Replication)

ج: مهاجرت (Migration) د: جابه جایی (Relocation)

۹. این نکته که یک تراکنش (Transaction) به صورت کامل انجام می گیرد یا انجام نمی گیرد، تحت عنوان کدام ویژگی شناخته می شود (فراگیر - ۸۸).

الف: یکپارچگی (Atomic) ب: ماندگاری (Durable)

ج: سازگاری (Consistency) د: انزوا (Isolation)

۱۰. کدام یک از موارد زیر، یک تعریف جامع در مورد سیستم های توزیع شده است؟ (فراگیر - ۸۹).

الف: مجموعه ای از کامپیوترهای مستقل که از دید کاربر یک سیستم یکپارچه می باشد.

ب: مجموعه ای از کامپیوترهای وابسته که از دید کاربر به شکل یک سیستم متمرکز دیده می شوند.

ج: مجموعه ای از کامپیوترهای وابسته که هر کدام بخش مشخص از فرآیند را انجام می دهند.

د: مجموعه ای از کامپیوترهای مستقل که از دید کاربر می توانند به شکل یک یا چند سیستم یکپارچه دیده شوند.

۱۱. مخفی ماندن حرکت یک منبع به مکانی دیگر در هنگام استفاده، یبائگر کدام نوع شفافیت (Transparency) در سیستم های توزیع شده است؟ (فراگیر - ۸۹ و ۹۰).

الف: دست یابی (Access) ب: مکان (Location)

ج: مهاجرت (Migration) د: جابه جایی (Relocation)

۱۲. شبکه های سنسوری چه نوع سیستم توزیع یافته می باشند؟ (فراگیر - ۸۹، ۹۰ و شهریور ۹۳).

الف: سیستم های فراگیر توزیع شده (Distributed Pervasive Systems)

ب: سیستم های شبکه ای توزیع شده (Distributed Network Systems)

ج: سیستم های اطلاعاتی توزیع شده (Distributed Information Systems)

د: سیستم های محاسباتی توزیع شده (Distributed Computing Systems)

۱۳. سیستم نام گذاری دامنه (DNS = Domain Name System) از چه تکنیک مقیاس پذیر (توسعه پذیری (Scaling)) بهره می گیرد؟ (فراگیر - ۸۹).

الف: کش کردن (Caching) ب: توسعه پذیری (Extensible)

ج: توزیع (Distribution) د: تکثیر (Replication)

۱۴. ویژگی های مهم، در توصیف یک سیستم توزیع یافته کدام است؟ (فراگیر - ۹۰).

الف: یک مجموعه از کامپیوترهای وابسته - مخفی ماندن این وابستگی از دید کاربر.

ب: مجموعه ای از کامپیوترهای مستقل - هم زمان بودن این کامپیوترها برای انجام یک یا چند فعالیت.

ج: یک مجموعه از کامپیوترهای مستقل - ظاهر شدن این مجموعه به شکل سیستم یکپارچه.

د: چندین کامپیوتر که می توانند با هم تبادل اطلاعات داشته باشند - مخفی ماندن تبادل اطلاعات از دید کاربر.

۱۵. تمام موارد زیر تکنیک های بسط پذیری (توسعه پذیری) (Scaling) محسوب می شود به جز: (فراگیر - ۹۰).

الف: کش کردن (Caching) ب: تکرار (Replication)

ج: توزیع (Distribution) د: ارتباط همگام (Synchronous Communication)

۱۶. کدام عبارت در مورد یک سیستم توزیع یافته صحیح نیست؟ (فراگیر - ۹۰).

الف: مجموعه‌ای از کامپیوترهای مستقل است.

ب: غالباً ارتباط از دید کاربر مخفی می‌باشد.

ج: جایگاه کامپیوتر در این نوع سیستم‌ها از دید کاربر مخفی می‌باشد.

د: دلیل توسعه‌پذیری این نوع سیستم‌ها، ظاهر شدن آن‌ها به شکل یک سیستم یکپارچه است.

۱۷. پنهان کردن حرکت یک منبع به موقعیت دیگر، توسط کدام نوع شفافیت (Transparency) توصیف می‌شود؟ (فراگیر - دی ۹۰ و شهریور ۹۳)

الف: مکان (Location) ب: مهاجرت (Migration)

ج: جابه‌جایی (Relocation) د: هم‌روندی (Concurrency)

۱۸. نقش میان‌افزار (Middleware) در یک سیستم توزیع‌شده چیست؟ (فراگیر - دی ۹۰)

الف: بهبود دادن توسعه‌پذیری سیستم ب: مشخص کردن نحوه ارتباط کامپیوترها با یکدیگر

ج: فراهم کردن دیده شدن سیستم به صورت یک سیستم واحد

د: ایجاد ارتباط بین سطح پایینی و بالایی در یک سیستم توزیع یافته

۱۹. کدام یک از موارد زیر، جز و تکنیک‌های توسعه‌پذیری (Scalability) در یک سیستم توزیع یافته نیست؟ (فراگیر - دی ۹۰).

الف: کش کردن (Caching) ب: تکرار (Replication)

ج: توزیع (Distribution) د: مخفی کردن ارتباط (Hide Communication)

۲۰. یک سیستم توزیع‌شده به سیستمی می‌گویند که از..... تشکیل شده باشد (فراگیر - ۹۱).

الف: سیستمی که تمام پردازنده‌هایش دارای معماری یکسانی باشند.

ب: یک سیستم با حافظه فیزیکی توزیع‌شده و حافظه منطقی مشترک.

ج: مجموعه‌ای از کامپیوترهای مستقل که برای کاربر همانند یک سیستم واحد و به هم چسبیده است.

د: یک سیستم موازی با حافظه مشترک

۲۱. مزیت شفافیت مکانی (Location Transparency) عبارت است از (فراگیر - ۹۲):

الف: ماهیت منابع را صرف‌نظر از این که در دیسک یا حافظه اصلی باشد، مخفی می‌کند.

ب: اجازه دسترسی به یک منبع بدون داشتن دانش مکان فیزیکی آن منبع را فراهم می‌کند.

ج: اجازه جابه‌جایی منابع و کاربران در سیستم بدون این که عملکرد کاربران یا برنامه‌ها تحت تأثیر قرار بگیرد را فراهم می‌کند.

د: اجازه دسترسی به منابع محلی و منابع راه دور (Remote) از طریق عملگرهای یکسان را می‌دهد.

۲۲. کدام گزینه جزء سیستم‌های چندین پردازنده‌ای است؟ (پیام نور - نیم‌سال اول ۹۰-۹۱).

الف: سیستم‌های چندپردازنده‌ای ب: سیستم‌های چند کامپیوتری ج: سیستم‌های توزیع‌شده

الف: گزینه‌های الف و ب ب: گزینه ب ج: گزینه‌های ج و ب د: موارد الف، ب و ج

۲۳. سیستم..... یک سیستم کامپیوتری است که از دو یا چندپردازنده تشکیل شده و آن‌ها دسترسی کامل به حافظه اشتراک RAM دارند (پیام نور - نیم‌سال اول ۹۰-۹۱ و نیم‌سال اول ۹۴-۹۵).