سیستم‌عامل پیشرفته

محمد داورپناه جزی

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی فولاد

ترم دوم ۹۴-۹۳

فصل دوم: ارتباطات در سیستم‌های توزیع شده

IPC در سیستم توزیع شده باید بدون استفاده از هیچ حافظه مشترکی انجام شود. (در سیستم متمرکز حتی سمافور یک حافظه مشترک است) یعنی بر مبنای انتقال پیغام و استفاده از مواردی چون RPC

مثال: تولیدکننده و مصرف‌کننده در سیستم متمرکز (کل موضوع IPC از اول)

1. پروتکل‌های لایه‌بندی شده: نحوهٔ ارسال پیام از A به B انگلیسی و ASC11 ⇔ به فرانسوی و Ebcdic

- فرستنده و گیرندهٔ پیام نیز به قراردادهایی در مورد پایین‌ترین سطح (ولتاژ بیت صفر و یک) تا بالاترین سطح نحوهٔ تفسیر اطلاعات هستند. به این قراردادها پروتکل گویند و از پایین‌ترین سطح (بین حافظه و CPU) تا بالاترین وجود دارند.
- قراردادها باید استاندارد باشند که ISO استانداردی به نام Open System Interconnechion Reference Model یا OSIارائه کرده است. (تعریف سیستم باز، دارای ارتباط مدون با سایر سیتمهای باز) ISOOSI. پروتکل بین انسان‌ها، بین CPU و حافظه
- پروتکل‌های ConnectionOriented و Connectionless (تلفن زدن در مقابل انداختن نامه)
- OSI دارای هفت طبقه یا لایه است، برای سهولت کار انجام شده، هر لایه سرویس به لایه قبلی می‌دهد. شکل ۱-۲ ص ۳۷

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- جهت ارسال پیغام M از A به B (شکل ۲-۲ ص ۳۸)، A پیغام را ساخته و در هر لایه یک Header به آن اضافه شده به لایه زیر می‌رود و (شاید یک Trailer هم) تا به شبکه برسد و در مسیر به طرف B در هر لایه Header مربوطه حذف می‌شود و از محتویات آن استفاده می‌شود .
  OSI از نوع
  Connection Oriented است.
- هر لایه مستقل از بقیه قابل تغییر و اصلاح است با پیشرفت تکنولوژی ، همین دلیل لایه بندی است.
- به مجموعه پروتکل‌ها Protcal Stack گویند.
- **لایه فیزیکی:** استاندارد ولتاژ، تعداد Pin، معنی هر کدام، یک طرفه یا دو طرفه بودن خط (صفر بدهیم و صفر بگیرد، یک بدهیم و یک بگیرد) مثال RS-232-C
- **لایه Data Link:** اطمینان از درستی Frame های بیت، افزودن Checksum در مبدأ و چک کردن آن در مقصد، در خواست ارسال مجدد در صورت وجود غلط، Frame ها شماره ردیف دارند. ش ۲-۳ ص ۳۹ نحوهٔ ارسال دو پیام و غلط بودن اولی

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **لایه Network: حل مسئله انتخاب مسیر ارسال با توجه به طول مسیر، ترافیک مسیرها و ... (routing) همیشه کوتاهترین مسیر بهترین نیست.**
- **X25 یک پروتکل Connection Oriented است است و پس از ایجاد ارتباط عمل انتقال را انجام می‌دهد.**
- **(Intenet Protocol) از نوع Connectionless است توسط DOD داده شده اکه در آن یک IPPacket بدون هیچ Setup اولیه ارسال می‌شود مستقل از بقیه بدون حفظ مسیر برای استفاده بعدی**
- **لایه Transport: تبدیل کل پیغام به Packet های با اندازه مناسب و شماره‌گذاری آنها، ارسال آنها**
- **ارسال می‌تواند به ترتیب (X25) یا بدون ترتیب (IP) باشد، در مقصد باید مجدداً مرتب گردند**
- **اطمینان از عدم گم شده‌گی یک Packet در همین لایه است.**
- **استاندارد DoD همان Transmisson Control Protocol یا TCP/IP که از نوع Connection Oriented است و UDP که از نوع دیگر است (Universal Data Gram Protocol) همان IP با کمی تغییر**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **لایه Session: مکمل لایه Transport است، معین می‌کند کی در حال حرف زدن است.**
- **کار دیگر این لایه ساختن Check Point در سول ارسال‌های طولانی است برای استفاده در توقف نابه‌جا**
- **در خیلی موارد لایه Sessionوجود ندارد (مثلاً در DoD)**

- **لایه Persentation: تعریف کردن فرمت اطلاعات در سطح بالا (ساختار رکورد) برای اطمینان از درستی ارسال و دریافت بخصوص وقتی دو طرف نمایش‌های داخلی متفاوت دارند. (دیگر با بیت کاری نداریم)**

- **لایه Application: پروتکل‌های عمومی مثل Email X400، FTP X500 دایرکتوری سرور، Remote Terminal و ... سه طبقه بالایی در این درس بحث نمی‌شود.**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

2. شبکه‌های ATM Asynchronous Transfer Mode Networks

- ARPANET سرعت 56Kbps.در سال 1969
- آخر دهۀ ۷۰ و اوائل ۸۰، نوع T1 با 1.5Mbps
- سپس T3 با 45Mbps
- نوع جدید با 155Mbps به سمت Gbps
- این تغییرات تأثیر بسیاری روی شبکه‌ها و سیستم‌های توزیع شده دارد. دانشگاه ما: 2 gbps

- **ATM:**
  - نیاز به انتقال Data در کنار خطوط تلفن برای صدا، شبکه تلفن فقط برای صدا نیست.
  - خطوط تلفن: عمل پیوسته ولی سرعت کم
  - خطوط Data: عمل انتقال ناپیوسته ولی حجم زیاد در فواصل زمانی کم
  - عدم کارآیی خطوط تلفنی قدیمی و خطوط جدید Internet برای هر دو مورد

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **راه حل ATM: ایجاد یک ارتباط از فرستنده به گیرنده(ها)، تعریف یک مسیر از فرستنده به گیرنده(ها)، ذخیره اطلاعات مسیر در سوئیچ‌های بین راه، تقسیم Packet ها به قطعات با طول ثابت به نام Cell و ارسال آنها با استفاده از اطلاعات مسیر ذخیره شده در سوئیچ‌ها، حذف ارتباط با پاک کردن اطلاعات مسیر از سوئیچ‌ها**
- **مزایـــا:**
  - **استفاده از یک شبکه برای ارسال انواع صدا، تصویر، Data، کنفرانس ... به جای چند شبکه متفاوت**
  - **وصل یک سیم به هر خانه برای تلفن، تلویزیون، کامپیوتر و ... (NYNEX)**
  - **امکان هر دو کار PointtoPoint (تلفن) و Multycasting (رادیو ـ تلویزیون)**
  - **تلفن امکان Multycast ندارد Cabletv امکان PointtoPoint دارد به قیمت از دست دادن امکانات زیاد**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- لایه‌های ATM Reference Model
  - لایه فیزیکی: یک برد ATM Adaptor متصل به کامپیوتر می‌تواند یک Stream ممتد از Cellها را به بیرون بفرستد. حتی وقتی اطلاعاتی ندارد Cell خالی بفرستد.پس در لایه فیزیکی Synchronus است.
  - یا می‌تواند از SONETیا (Synchronous Optical Network) استفاده کند و Cellها را در بخش اصلی (Payload) از Frame های آن قرار دهد.
  - یک فریم SONET آرایه 90=810*90 بایت (۳۶ بایت بالا سری و ۷۷۴بایت Payload) است و در 125µsec منتقل می‌شود 8×810=6480 بیت و در یک ثانیه می شود 8000×6280=51840000 ناخالص و 49536 خالص
  - سرعت Mbps 51.840 است که یک کانال را OC1 و n تا جدا را OCn و با هم را OCnc گویند.
  - استانداردها: OC-3 ، OC-3c Concatented155.520 (کامپیوتر)، OC-12،12 (622.08mBPS) OC-12c (کامپیوتر)، OC-48 و OC-192

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **لایه‌های ATM**
  - **در اروپا سلول ۳۲ بایتی می‌خواستند و در آمریکا سلول ۶۴ بایتی که روی ۴۸ بایتی + ۵ بایت Header توافق شد که برای صدا زیاد بود و برای Data کم بود و هیچکس را راضی نکرد**
  - **۷۷۴ بایت سونت هم مضربی از آن نمی‌شد پس باید روی سونت Span می‌شد.**
  - **دو مرحله سنکرونیزاسیون نیاز بود یکی برای شروع فریم و دیگری شروع اولین سلول کامل**
  - **استانداردی برای این تعریف شده که به طور کامل در سخت‌افزار انجام می‌شود.**
  - **فرمت Header از کامپیوتر به سوئیچ ATM شکل ۵-۲ ص ۴۶**
  - **از ATM sw. به ATM sw. فیلد اول سمت چپ GFC به ادامه VPI تبدیل می‌شود. ولی کنترل فلو است برای آینده**
  - **VPI و VCI معین‌کننده نوع اطلاعات اصلی (کنترلی یا Data) است و نوع کنترل**
  - **CLP نماینده سلول‌های کم اهمیت است که در صورت شلوغی شبکه می‌تواند حذف شود.**
  - **فیلد آخر هم Checksum است فقط روی Header**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **لایه ATM Adaptation**
  - **با توجه به سرعت 155Mbps، هر سلول در سه میکرثانیه [424=8×(48+5)]یا [155×3 وارد می‌شود یعنی در یک ثانیه نیاز به 300,000 تا وقفه داریم که کمتر CPUای می‌تواند این کار را بکند**
  - **یک مکانیزم تبدیل Packet که کامپیوتر می‌دهد به سلول برای انتقال و برعکس نیاز داریم که بر مبنای یک وقفه در هر Packet کار کند نه هر سلول.**
  - **این لایه چنین کاری را انجام می دهد. (شکستن Packet به سلول و برعکس)**
  - **در اینجا هم نویسنده استاندار درست عمل نکرده و در آغاز ۴ کلاس در ص ۴۷ تعریف شد.**
  - **دلیل آن هم مشکل بودن استانداردسازی بین دو تکنولوژی تلفن و کامپیوتر است.**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- کامپیوتری‌های بیدار شدند و متوجه عدم کارآیی این استانداردها گردیدند و AAL5 را پیشنهاد کردند.
- SEAL یا Simple & Efficient Adaptation Layer اسم مستعار آن
- در SEAL یک بیت از Pagload Type که معمولاً 0 است در آخرین سلول یک Packet به 1 تبدیل می‌شود.
- بعد از آن هم Padding و سپس Trailer قرار دارد که در مقصد مورد استفاده قرار می‌گیرد و برای پایان Packet
- Trailer هم حاوی اطلاعاتی در مورد طول Packet و Checksum آن است (کلاً ۸ بایت)
- روی همه لایه‌ها، لایه‌هایی برای عملیات سطح بالاتر نیاز است مثلاً تبدیل سلول‌ها به TCP/IP

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **ATM Switching**
  - **شکل ۶-۲ ص ۴۸ شبکه ATM و داخل یک سوئیچ را نشان می‌دهد.**
  - **سلول‌ها از هر کامپیوتر به هر کامپیوتر دیگر از طریق سوئیچ‌ها منتقل می‌شود.**
  - **هر SW دارای چهار خط ورودی / خروجی همزمان و یک Switching Fabric موازی برای انتقال آنها دارد.**
  - **هر سلول در OC-3 در سه µs وارد می‌شود و چهار تا هم همزمان می‌تواند باشد پس موازی بودن اساسی دارد.**
  - **با ورود یک سلول از طریق مقایسه VCI و VPI آن با مقادیر ذخیره شده در سوئیچ (در زمان تعریف مسیر) آن سلول به خروجی مورد نظر رد می‌شود.**
  - **اگر دو سلول با هم برسند و به مقصد یک خروجی باشند: یا یکی حذف می‌شود گرچه استاندارد است ولی خوب نیست**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- می‌توان یکی را ارسال و دیگری را Queue کرد در ورودی که باعث بلوکه شدن سلول‌های بعدی می‌شود. به این حالت Head of Line Blocking گویند.
- می‌توان صف را در خروجی تشکیل داد که عیب بالارا ندارد.
- می‌توان بافرهایی داشت مشترک بین ورودی و خروجی
- راه حل‌های دیگر هم هست.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **آثار ATM روی سیستم‌های توزیع شده**
  - **ظرفیت 155Mbps و بالقوره 2.5Gbps یک باند بسیار وسیع را ایجاد می‌کند برای سیستم‌های توزیعی Widearea**
  - **در ارسال یک فایل 1Mbit در عرض آمریکا 15ms زمان انتقال یک بیت است (۲/۳ سرعت نور در خلاء برای ۳۰۰۰ کیلومتر سیم‌های مسی یا نوری)**
  - **در 64Kbps عمل ارسال 15.6 ثانیه طول می‌کشد که 30ms تأثیری روی آن ندارد (رفت و برگشت)**
  - **در 62Mbps عمل ارسال 1.6ms است جمع رفت و برگشت 31.6ms می‌شود که در 30ms (95%) آن خط ارسال بیکار است.**
  - **هر چه سرعت بالا رود مدت کارکرد عمل ارسال به سمت صفر میل می‌کند (خصوصاً فایل‌های کوچکتر)**
  - **پس برای سیستم‌های توزیع شده در این شرایط نیاز به پروتکل و معماری جدید است.**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- مسئله دیگر هماهنگی در کنترل جریان اطلاعات است.
- در ارسال یک ویدئو ده GB گیرنده پیغام STOP به دلیل کمبود جا بدهد
- تا پیغام برسد به فرستنده ۳۰ میلی ثانیه، 20Mb در راه است که باید دوباره ارسال شود (سرعت 622Mbps)
- روش پنجره لغزان 1Mb بفرستد و صبر کند تا برسد می‌شود همان 95% بیکاری دستگاه
- می‌توان بافرهای حجیم ایجاد کرد که هزینه زیاد دارد.
- امکان دیگر توافق قبلی فرستنده و گیرنده روی نرخ انتقال است.
- پیشنهاد دیگر ارسال مقطعی است یعنی مدتی ارسال بعد توقف و پرداختن بکارهای دیگر تا جواب بیاید.
- ارزانی CPU باعث شده که هر Process روی یک CPU مخصوص بخود اجرا شود پس کار دیگری برای اجرا نیست.
- این روزها هدر دادن زمان CPU آنقدرها اهمیت ندارد!

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

پس : در وضعیت فعلی رفتن از 64Kbps به 622Mbps، کارآیی سیستم‌ها را ده‌هزار برابر نکرده است.

- **مثال برنامه‌ای در NY تعداد ۲۰درخواست ردیفی به CA بدهد که 600ms تأخیر دارد که قابل ملاحظه است.**
- **راه دیگر انتقال برنامه به CA و اجرا در آنجاست و ارسال هر کاراکتر مستقلاً با 60ms تأخیر که قابل قبول است.**
- **این دیگر سیستم توزیع شده نیست بلکه یک سیستم بزرگ مرکزی با کاربران توزیع شده است.**
- **خصوصیت دیگر حق Drop کردن سلول‌هاست در صورت شلوغی شبکه، که می‌تواند به ارسال مجدد کل Packet ها بیاجامد.**

نتیجه: گر چه شبکه‌های سریع نظیر ATM امکانات بسیار قوی دارند ولی استفاده مؤثر از آنها ساده نیست. هنوز راه زیادی در تحقیقات باید طی شود.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

3. مدل Client Server

- به دلیل گرانی فعلاً امکان استفاده از ATM در همه جا وجود ندارد پس برگردیم به راه‌حل‌های معمول‌تر
- روش OSI برای استفاده در سیستم توزیع شده به نظر عملی می‌رسد (آیا اشکالی دارد؟)
- وجود تعداد زیادی لایه‌های پروتکل در دو طرف و آن همه Header چسبیده به هر پیغام
- زمان لازم برای بررسی و تطبیق آنها زیاد است، عدم استفاده از پروتکل‌ها در LAN یا استفاده از بخش کوچکی شکل ۲-۷ ص ۵۱
- در شبکه‌های Widearea که سرعت انتقال کم است اشکالی ندارد.
- فاکتور محدودکننده ظرفیت خط است و با وجود همهٔ موارد فوق سرعت بالای CPUها از عهدهٔ کار پردازش Headerها برمی‌آید.
- ولی در سیستم توزیع شده بر مبنای LAN هزینه بالا سری پروتکل‌ها زیاد است.
- به همین دلیل در اکثر LAN ها یا کلاً از این پروتکل‌ها استفاده نمی‌شود و یا از بخش کوچکی از آن استفاده می‌گردد.
- مدل OSI فقط بخش خاصی از انتقال بیت‌ها از فرستنده به گیرنده را حل می‌کند . ساختار سیستم توزیعی را معین نمی‌کند.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **Clinet ها و Server ها شکل ۷-۲ ص ۵۱**
  - **ساختار دادن سیستم عامل در قالب تعدادی Process همکار سرویس‌دهنده که درخواست‌های کاربران یا سرویس گیرنده‌ها را در قالب سرویس‌های خود انجام می‌دهند.**
  - **ماشین‌های هر دو گروه دارای یک Micro Kernel هستند و به عنوان پردازش‌های user کار می‌کنند.**
  - **یک ماشین می‌تواند یک پردازش تک را انجام دهد یا تعدادی پردازش‌ها از هر گروه یا مخلوطی از آنها**
  - **برای جلوگیری از Overhead به جای OSI یا TCP/IP از پروتکل‌های Connetionless از نوع Requst/Reply استفاده می‌کند.**
  - **Cilent پیغامی مبنی بر درخواست یک سرویس می‌فرستد.**
  - **Server یا نتیجه را می‌فرستد یا یک کد خطا ش ۹-۲، مهمترین خصوصیت سادگی کار است و کارآیی بیشتر**
  - **نیاز به ایجاد Connection قبل از عملیات نیست.**
  - **طبقات پروتکل ۳ تا ست (شکل b7-2)**
  - **سطح فیزیکی و دیتا لینک توسط سخت‌افزار عمل انتقال بیت‌ها را انجام می‌دهد**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- سطح ۳ و ۴ لازم نیست چون Routing و Connectionای وجود ندارد.
- در سطح ۵ مجموعه درخواست‌ها و جواب های قانونی و قابل قبول تعریف می‌شود.
- سطح Session هم وجود ندارد و لایه‌های قبلی هم نیازی نیست.
- با توجه به سادگی در Micro Kernel فقط دو احضار سیستم نیاز داریم (درس .(IE

  Send (Dest, & mptr) Receive (Address ,& mptr)
- sendارسال پیام به پروسس مقصد و بلوکه کردن احضار کننده تا تکمیل ارسال
- receiveبلوکه کردن احضار کننده تا وصول پیام
- مثال Cilent/Server شکل ۸-۲ و ۹-۲ ص ۵۳ و ۵۵
- در Server (بالا) اول Receive با دو پارامتر اولی شماره Server و دومی آدرس بافر احضار می‌شود تا وقتی پیامی نیامده برنامه Server بلوکه می‌شود و وقتی آمد ادامه می‌یابد (TrapstoKernel) بعد از تعیین نوع کار و انجام آن جواب را می‌فرستد و تکرار حلقه.
- در Client پس از آماده‌سازی، پیامی می‌رود برای خواندن و نتیجه دریافت می‌شود و بعد نتیجه روی فایل دیگر با استفاده از دو پیام نوشته می‌شود.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- **آدرس‌دهی**
- **ارسال شماره ماشین به همراه پیام، دریافت آن فقط توسط ماشین مربوط ، در صورت وجود یک پروسس روی هر ماشین**
- **ارسال شماره پروسس + شماره ماشین، وصول پیام در ماشین مربوطه از روی شماره ماشین، تحویل به پروسس مناسب از روی شماره پروسس**
- **شماره پروسس می‌تواند ردیفی باشد یا عدد تصادفی (Local-id) شکل ۲-۱۰a ص ۵۷**
- **مثلاً در Berkeley Unix سرویس‌دهنده با دادن Localid خود به Kernel منتظر وصول پیام می‌ماند**

هنگام وصول پیام توسط Kernel، می‌داند که پیام را به کی بدهد. (چون پروسس مربوطه گوش می‌دهد به LTP)

- **شماره پروسس + شماره ماشین به دلیل Transparent نبودن برای کاربر، روش مناسبی نیست**
- **اگر File Server شماره ۲۴۳ باشد و خراب شود ماشین دیگری نمی‌تواند جایش را بگیرد (نیاز به اصلاح ۲۴۳ در Header)**

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- راه دیگر دادن شماره ردیف یگانه به همه پروسس‌ها از طریق یک پروسس مرکزی است، در سیستم‌های بزرگ کارآیی ندارد (عنصر متمرکز)
- راه سوم تخصیص شماره تصادفی از یک فاصله بزرگ (مثلاً ۶۴ بیت) است شبیه Hash
- در این روش شناسایی ماشین اگر در LAN باشد از طریق روش Boardcast انجام می‌شود شکل ۲-۱۰b ص ۵۷
- فرستنده با Locate Packet شماره را روی شبکه می‌فرستد ماشین مربوطه جواب می‌دهد با شماره خودش، فرستنده شماره را حفظ می‌کند (درس IE).
- عیب این روش عملیات اضافی است (خلاصه در ص ۵۸)
- روش دیگر استفاده از Name Server برای تبدیل اسامی سمبولیک به شماره منطقی است که باز هم یک عنصر متمرکز است (درس IE).
- راه آخر ذخیره شماره پروسس‌ها در سخت‌افزارهای سوئیچینگ است.
- با رسیدن Packet به Network Interface Chip سوئیچ شماره پروسس را با اطلاعات خودش مقایسه می‌کند و اگر داشت آنرا به ماشین مربوطه میفرستد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه) اینجا

- دستورات پایه‌ای با بلوک کردن و بدون بلوکه کردن، **Synchronous** در مقابل **Asynchroneus**
- در **Send** و **Receive** بلوکه کننده، پروسس احضارکننده تا خاتمه انجام کار بلوکه می‌شود شکل ۲-۱۱۹ ص ۵۹
- در روش بدون بلوکه کردن کنترل به احضارکننده برمی‌گردد ولی امکان استفاده از بافر را ندارد (نمی‌داند کی خالی می‌شود)
- راه حل اول کپی کردن بافر در **Kernel** است شکل ط ۲-۱۱ ص ۹۵ که زمان اضافه نیاز دارد.
- راه حل دوم وقفه دادن به احضارکننده است وقتی بافر خالی شد که برنامه‌نویسی آن پیچیده است (همان داستان کفش و کلاه یا مقنعه!).
- حالت اول در شرایط معمولی با وجود سطح پایین توازی، قابل قبول است چون پیاده‌سازی آن ساده است و بدون کپی کردن پیغام سریعتر آماده انتقال است (خلاصه ص ۶۰)
- حالت دوم وقتی خوب است که **Overlapping** بین پردازش و انتقال حیاتی باشد.
- **Receive** بدون بلوکه آدرس را به **Kernel** می‌دهد و برمی‌گردد. برای اطلاع از خاتمه کار یا خودش باید دستور **Wait** داشته باشد یا **Kernel** را **Poll** کند یا اینکه نهایتاً از **Internept** استفاده شود.
- در کل **Receive** با بلوکه کردن ترجیح داده می‌شود.
- موضوع دیگر نیاز به **Timeout** است وقتی بلوکه کردن طول کشید که با یک کد **Error** باید برگردد.

فصل دوم: ارتباطات در سیستم‌های توزیع شده (ادامه)

- دستورات پایه‌ای با و بدون بافرینگ
  - حالتی که در بالا بحث شد بدون بافر هستند.
  - یک (Receive (Add , &mبه آدرس یک پروسس در Server اشاره کرده و منتظر ارسال پیغام از آن است شکل ۲-۱۲a ص ۶۲
  - وقتی درست کار می‌کند که (Receiver قبل از (...) Send اتفاق بیفتد
  - اگر اینطور نبود چه می‌شود؟ Kernel گیرنده پیغام نمی‌داند آنرا به چه Processای بدهد!
  - یک راه صرفنظر و امید به ارسال مجدد است که می‌تواند به دلیل تکرار منجر به Timeout ارسال‌کننده شود.
  - در حالت چند Client اولی که درخواستش رسید و پذیرفته شد مال بقیه رد می‌شود (فکر می‌کنند Server خراب است)
  - در بافر کردن پیغام‌ها که وارد می‌شوند و با اسم Process تطبیق دارند وارد Meilbox می‌شوند.
  - هنگام نیاز، پیغام‌ها از آنجا برداشته می‌شوند و یا پروسس گیرنده بلوکه می‌شود اگر BOX خالی باشد.
  - اشکال وقتی است که BOX پر شد همان وضعیت قبلی پیش می‌آید پس مسئله تخفیف یافته ولی حل نشده
  - می‌توان پروسس فرستنده را در صورت پر بودن BOX معلق کرد (Block شود تا ack بدستش برسد)