۳۹) ارتباط مستقیم - غیر مستقیم (در سیستم تبادل پیام)

نام‌گذاری فرآیندها: فرآیندها برای ارتباط با یکدیگر نیازمند روشی برای ارجاع به یکدیگر هستند.

نحوه ارتباط فرآیندها با یکدیگر می‌تواند به صورت مستقیم یا غیر مستقیم صورت گیرد.

۴۰) ارتباط مستقیم (Direct communication)

در ارتباط مستقیم می‌بایست به طور صریح نام دریافت کننده و فرستنده مشخص باشد.

بنابراین در این طرح اولیه‌های (Primitives) عملیاتی Send و Receive به صورت زیر تعریف می‌شود:

- ارسال پیام به فرآیند P_1 : Send(P_1, message)

- دریافت پیام از فرآیند P_2 : Receive(P_2, message)

یک پیوند ارتباطی (Communication Link) در طرح تبادل پیام یا ارتباط مستقیم دارای ویژگی‌های زیر است:

۱- پیوند (Link) به صورت خودکار بین هر زوج فرآیندی که خواهان ارتباط هستند برقرار می‌شود. بنابراین فرآیندها می‌بایست از شناسه یکدیگر برای ارتباط مطلع باشند.

۲- یک پیوند ارتباطی دقیقاً برای دو فرآیند در نظر گرفته می‌شود.

۳- بین هر زوج فرآیند، دقیقاً یک پیوند وجود دارد.

۴۱) طرح تبادل پیام با ارتباط مستقیم از نظر آدرس‌دهی دارای دو رده کلی است:

- طرح متقارن،
- طرح نامتقارن.

(۴۲) طرح آدرس دهی متقارن: در طرح آدرس دهی متقارن، فرآیند فرستنده و فرآیند دریافت کننده برای برقراری ارتباط ملزم بایست از نام یکدیگر مطلع باشند. در واقع اولیه های عملیاتی همانطور که پیش تر نیز بیان شد به صورت زیر تعریف می شود.

- ارسال پیام به فرآیند P1: Send(P1, message)

- دریافت پیام از فرآیند P2: Receive(P2, message)

(۴۳) طرح آدرس دهی نامتقارن: در طرح آدرس دهی نامتقارن، تنها فرآیند فرستنده نیاز به استفاده از نام گیرنده دارد و گیرنده نیاز به نام فرستنده ندارد. اولیه های عملیاتی این طرح به صورت زیر است.

- ارسال پیام به فرآیند P1: Send (P1, message)

- دریافت پیام از هر فرآیندی: Receive (id, message)
جایگزین متغیر id با نام فرآیندی که در ارتباط شرکت می کند، تنظیم می شود.

نکته: پارامتر اول تابع Receive در طرح متقارن ثابت (constant) است و در طرح نامتقارن متغیر (Variable) است.

(۴۴) عیب طرح آدرس دهی متقارن / نامتقارن: در هر دو طرح نیاز است به هنگام استفاده از اولیه های عملیاتی send و Receive، شناسه فرآیندها به صراحت بیان شود. بنابراین تغییر پارامترهای این اولیه ها سبب آزمون تمامی تعاریف فرآیندها را در پی خواهد داشت. به طور کلی در تکنیک Hard-coding که در آن شناسه ها باید به صراحت بیان شوند، مطلوبیت کمتری نسبت به تکنیک های غیر مستقیم دارند.

(۴۵) ارتباط غیرمستقیم (Indirect communication)

در این روش، پیام ها از طریق صندوق پستی (Mailbox) یا پورت (Port) ارسال و دریافت می شوند.

یک صندوق پستی به طور انتزاعی شیئی است که فرآیندها می توانند پیام در آن قرار دادن یا پیام موجودی را حذف کنند. لازم به ذکر است که هر صندوق پستی یک شناسه یکتا دارد.

مثال صندوق پستی: برای نمونه صف پیام POSIX یک مقدار صحیح را به عنوان شناسه یکتا برای صندوق پستی در نظر می گیرد. یک فرآیند می تواند از طریق تعدادی صندوق پستی با دیگر فرآیندها ارتباط برقرار کند. در این طرح، اولیه های عملیاتی send و receive به صورت زیر تعریف می شوند:

- ارسال پیام به صندوق پستی با شناسه A: send(A, message)
- دریافت پیام از صندوق پستی با شناسه A: receive(A, message)

(۴۶) یک پیوند ارتباطی (Communication Link) در طرح تبادل پیام یا ارتباط غیر مستقیم دارای ویژگی های زیر است:

۱- یک پیوند (Link) بین یک زوج فرآیند ایجاد می شود اگر هر دو عضو این زوج فرآیند، صندوق پستی مشترک داشته باشند.

۲- یک پیوند (Link) می تواند به بیش از دو فرآیند اختصاص داشته باشد.

۳- بین هر زوج فرآیندی تواند بیش از یک پیوند (هر پیوند منطبق با یک صندوق پستی) وجود داشته باشد.

(۴۷) سؤال:

فرض کنید فرآیندهای P_1، P_2 و P_3 همگی صندوق پستی A را به اشتراک گذاشته و عملیات زیر انجام می دهند:

* فرآیند P_1 پیام m را به صندوق پستی A ارسال می کند،
* فرآیند P_2 و P_3 عمل Receive را به طور همزمان اجرا می کنند،

حال صورت سؤال آن است که کدام فرآیند پیام ارسالی توسط فرآیند P_1 را دریافت می کند؟

پاسخ: پاسخ به این سؤال وابسته به روش های انتخابی زیر دارد:

* اجازه اختصاص یک پیوند به حداکثر دو فرآیند،
* اجازه حداکثر یک فرآیند برای اجرای عمل Receive،
* اجازه به سیستم به انتخاب اختیاری فرآیند دریافت کننده پیام (به عنوان مثال فرآیند P_2، فرآیند P_3 یا هیچکدام.

در رویکرد سوم در واقع الگوریتمی برای انتخاب فرآیند هدایی و در نظر گرفته می شود به عنوان مثال الگوریتم نوبت گردشی (Round-Rabin) که در آن فرآیندها به نوبت و به صورت چرخشی، پیام ها را دریافت می کنند.

(۴۸) مالکیت صندوق پستی

مالکیت صندوق پستی به دو صورت زیر مالکیت می شود:

۱- توسط یک فرآیند مشارکت کننده در ارتباط و به عنوان بخشی از مفهوم آدرس آن،

۲- توسط یک فرآیند مستقل ایجاد شده توسط سیستم عامل.

* در حالت اول (مالکیت توسط فرآیند مشارکت کننده در ارتباط)، صندوق پستی به عنوان بخشی از فضای آدرس فرآیند در نظر گرفته می‌شود. در این حالت، مالک صندوق پستی تنها می‌تواند پیام‌های ارسالی را دریافت و فرآیندهای دیگر تنها می‌توانند اقدام به ارسال پیام به صندوق پستی مذکور داشته باشند.

لازم به ذکر است که از آنجایی که هر صندوق پستی یک مالک یکتا دارد، ابهامی در رابطه با اینکه کدام فرآیند دریافت کننده پیام ارسالی است، وجود ندارد. همچنین هنگامی که فرآیند مالک یک صندوق پستی خاتمه می‌یابد، صندوق پستی حذف می‌شود و می‌بایست هر فرآیندی که پیامی را متعاقباً به صندوق پستی مذکور می‌فرستد از عدم وجود آن مطلع شود.

* در حالت دوم (مالکیت توسط فرآیند مستقل ایجاد شده توسط سیستم عامل)، صندوق پستی به صورت مستقل از فرآیندهای مشارکت کننده در ارتباط مالکیت می‌شود. در این حالت سیستم عامل می‌بایست مکانیزمی را در این رابطه مهیا کند که:

1. امکان ایجاد صندوق پستی جدید وجود داشته باشد،
2. امکان ارسال (Send) و دریافت (Receive) پیام از طریق صندوق پستی وجود داشته باشد،
3. امکان حذف صندوق پستی وجود داشته باشد.

در این حالت:

- فرآیندی که خالق صندوق پستی است به طور پیش فرض مالک آن است.
- به طور ابتدایی، مالک صندوق پستی تنها فرآیندی است که می‌تواند پیام‌های ارسالی به صندوق پستی را دریافت نماید.

- مالکیت صندوق پستی و الویت دریافت پیام می‌تواند با استفاده از فراخوان‌های نسبتی به دیگر فرآیندها اعطا شود.

۴۹ همگام‌سازی (synchronization)

همانطور که پیش‌تر صحبت کردیم ارتباط فرآیندها از طریق تبادل پیام با استفاده از اولیه‌های send و Receive انجام می‌شود. با اینحال گزینه‌های طراحی متفاوتی برای پیاده‌سازی هریک از اولیه‌های فوق وجود دارد:

* بلاک کردن (Blocking) یا همگام (synchronous)
* بلاک نکردن (Nonblocking) یا غیرهمگام (Asynchronous).

بنابراین می‌توان با بهره‌گیری از گزینه‌های فوق و اولیه‌های send و Receive چهار حالت زیر را متصور شد:

1. Blocking send: فرآیند ارسال کننده تا زمانیکه پیام توسط فرآیند دریافت کننده یا صندوق پستی دریافت شود، بلاک می‌شود.
2. Nonblocking Send: فرآیند ارسال کننده پیام را ارسال و بدون توقف، عملیات خود را ادامه می‌دهد.
3. Blocking Receive: فرآیند دریافت کننده تا زمانیکه پیامی را دریافت کند، بلاک می‌شود.
4. Nonblocking Receive: فرآیند دریافت کننده یا یک پیام معتبر یا یک پیام تهی (NULL) بازیابی می‌کند.

۵۰ مطلب تکمیلی: می‌توان ترکیبی از گزینه‌های بالا را برای Send و Receive در نظر گرفت. در این رابطه اگر هر دوی Send و Receive از نوع Blocking باشد اصطلاحاً می‌گوییم یک قرار ملاقات (Rendezvous) بین گیرنده و فرستنده وجود دارد.

مسأله تولیدکننده-مصرف‌کننده در این حالت بدیهی است. تولیدکننده صرفاً blocking send را فراخوانی کرده و منتظر می‌ماند تا اینکه پیام تحویل صندوق پستی یا دریافت‌کننده شود. به طور مشابه وقتی مصرف‌کننده blocking Receive را فراخوانی می‌کند، فرآیند مصرف‌کننده تا زمانیکه پیامی قابل دسترس شود، بلاک می‌شود.

۵۱ بافر کردن (Buffering)

خواه ارتباط مستقیم باشد یا غیرمستقیم، پیام‌های تبادل شده توسط فرآیندهای مشارکت‌کننده در ارتباط به طور معمول در صف موقتی قرار می‌گیرند. چنین صف‌هایی معمولاً به سه طریق زیر پیاده‌سازی می‌شوند:

۱) ظرفیت صفر (zero capacity): در این روش، ماکزیمم طول صف صفر است بنابراین هیچ پیام در حال انتظاری نمی‌تواند در پیوند (Link) وجود داشته باشد. در این روش، فرستنده تا زمانیکه دریافت‌کننده پیام را دریافت کند، بلاک می‌شود.

۲) ظرفیت محدود (Bounded capacity): در این روش، ظرفیت صف محدود بوده و حداکثر n پیام می‌تواند در آن جای دهی شود. وقتی

یک پیام جدید ارسال می‌شود در صورتیکه:

* صف پر نباشد، پیام در صف قرار می‌گیرد (یا پیام در صف کپی می‌شود یا اشاره‌گر به آن در صف قرار می‌گیرد) و فرستنده بدون توقف به اجرای خود ادامه می‌دهد.

* صف پر باشد، فرستنده تا زمانیکه صف فضای آزاد پیدا کند، بلاک می‌شود.

③ ظرفیت نامحدود (unbounded capacity): ظرفیت صف به طور بالقوه نامتناهی فرض می‌شود. بنابراین هر تعداد پیام می‌تواند در آن منتظر بماند و هرگز بلاک نمی‌شود.

(۵۲) مطالب تکمیلی: روش Zero-capacity گاهی اوقات به عنوان سیستم تبادل پیام بدون بافر و روش‌های Bounded-capacity و Unbounded-capacity به عنوان بافر خودکار شناخته می‌شوند.