

CORBA

استاندارد ساختار معماری CORBA توسط گروه OMG و به منظور برقراری ارتباط بین اشیاء توزیع شده تدوین گردیده است. اشیائی که خود حاوی ماجولهای نرم افزاری می باشند.

CORBA: Common Object Request Broker Architecture

OMG: Object Management Group

CORBA شیوه‌های را عرضه نموده است که از طریق آن اجرای برنامه های (اشیاء) نوشته شده به زبانهای مختلف صرفنظر از سخت افزار و محل قرارگرفتن آنها در شبکه، امکانپذیر می گردد. این استاندارد برای معماری‌های چندلایه ای و برنامه های کاربردی client/server بویژه در مواقعی که اجرای یک فرآیند به اجزای فرآیندی بر روی کامپیوتر دیگری بستگی دارد مناسب می باشد.

برخی مواقع CORBA را به عنوان یک (مسیر عمومی نرم افزاری) و یا (مسیر اشیاء) توصیف می نمایند. زیرا درواقع یک واسط ارتباطی مبتنی بر نرم افزار است که از طریق آن مکان اشیاء و دسترسی به آنها انجام می پذیرد استاندارد CORBA درواقع مؤلفه ارتباطات معماری مدیریت اشیاء (OMA-Object Management Architecture) می باشد

از طریق CORBA سایر المانها نظیر security services ، naming services و transaction services معرفی می گردند.

اغلب عبارت CORBA را بجای OMA مورد استفاده قرار می دهند. اشیاء CORBA توسط زبان IDL تعریف می شوند. این زبان فرآیندها (متدهای) شیء را شرح داده و فرمت داده های ارسالی و مقادیر برگشتی را معین می نماید. برای هر یک از زبانهای برنامه نویسی مانند Smalltalk ، Java ، C++ ، C و COBOL یک کامپایلر مجزا در زبان IDL در نظر گرفته شده است. بدین ترتیب این امکان برای برنامه نویسان فراهم گردیده است تا متدهای سازنده را بکار برند که در سایر زبانها با آنها آشنا می باشند و برنامه های کاربردی client و server هر کدام با زبان مربوط به خود صحبت نمایند.

IDL: Interface Definition Language

اشیاء تعریف شده توسط IDL در Repository واسط ذخیره می گردند به نحوی که اشیاء ذخیره شده می توانند توسط برنامه های کاربردی client به منظور آگاهی از اینکه چه اشیائی، در مسیر (bus) قابل دستیابی می باشند، مورد استعلام قرار گیرند.

هرگاه در زمان اجرا درخواستی توسط یک client مربوط به شیئی که در مکان دوری بر روی شبکه ارتباطی قرار دارد، صادر گردد، ORB-Object Request Broker این پیگیری را انجام می دهد.

معماری CORBA

اجزای مختلف معماری corba بشرح زیر است:

- ۱- ORB یا واسط تقاضای اشیاء که یکی از قسمتهای اساسی معماری corba است.
- ۲- IDL یا زبان تعریف واسط که یکی دیگر از قسمتهای اساسی معماری corba است.
- ۳- مدل اشیاء corba که شامل ارجاعات اشیاء و سازگار کننده پایه ای اشیاء (BOA) است.
- ۴- مدل ارتباطی corba یعنی اشیاء corba چگونه در داخل یک شبکه با هم هماهنگ می شوند.
- ۵- تعریف و نقش مشتری و خدمتگذار در معماری CORBA

۱- دلال تقاضای شیئی ORB-object request broker

مفهوم یک ORB چنین شرح داده میشود: وقتی یکی از اجزا یکی از نرم افزارها بخواهد از سرویسهای فراهم شده توسط شیئی دیگری استفاده کند. ابتدا باید یک ارجاع از شیئی که سرویس را فراهم می کند بدست آورد. بعد اینکه این اشاره گر را بدست آورد آن قسمت میتواند متدهای فراهم شده توسط آن شیئی را فراخوانی کند و به سرویس مورد نظر خود برسد.

وظیفه اصلی ORB تصمیم گیری در مورد مراجعه به شیئی مورد نظر است و همچنین نرم افزارها را قادر می کند تا با یکدیگر ارتباط برقرار کنند. این مهمترین مسئولیت ORB است.

مارشالینگ

بعد از اینکه یک جزء نرم افزاری از شیئی که می خواهد از متدهای آن استفاده یک ارجاع کسب کرد، میتواند متد مورد نظر را فراخوانی کند این متدها پارامترهایی را به عنوان ورودی می گیرد و همچنین پارامترهایی را به عنوان خروجی می فرستد مسئولیت دیگر ORB دریافت پارامترهای ورودی از جزء فراخواننده و مارشال کردن آنها به شیئی فراهم کننده متد است همچنین پارامترهای خروجی از متدها را آنمارشال می کند.

مارشالینگ پروسه ای است که پارامترهای ورودی را به شکلی که قابل انتقال توسط شبکه باشد تبدیل می کند.

آنمارشالینگ: عمل معکوس مارشالینگ است و داده های آمده از شبکه را به شکل پارامترهای ورودی تبدیل می کند.

برای اینکه ORB بتواند داده ها را در شبکه رد و بدل کند آنها را به فرمت خاصی تبدیل می کند که این فرمت را one-the-wire می گویند

نتیجه مارشالینگ و آنمارشالینگ استقلال سخت افزاری است. بدلیل اینکه پارامترها هنگام انتقال از شبکه به یک شکل مستقل از سخت افزار تبدیل می شود و هنگام دریافت به یک شکل وابسته به سخت افزار تبدیل می گردد. در نتیجه ارتباط بین اجزاء یک رابطه مستقل از سخت افزار است. برای مثال یک سیستم مکینتاش می تواند متدی را در سروری که یونیکس است اجرا کند. تمام اعمال مارشالینگ و آنمارشالینگ توسط ORB اداره می شود.

ORB دارای چند بخش می باشد که مهمترین آنها موارد زیر هست:

1- انبار واسط

انبار واسط (IR) یک بخش خیلی مهم از ORB است که شامل اطلاعات در مورد نوع واسط است و نیازمند نوعی از حافظه دائمی برای نگهداری و ذخیره واسط است. اطلاعات موجود در فایل IDL برای ذخیره شدن در IR مناسب هستند.

2- انبار پیاده سازی

این انبار شامل اطلاعاتی در مورد موقعیت مکانی اشیاء و همچنین اطلاعاتی در مورد اینکه کدام یک از اشیاء اخیراً نمونه سازی شده است را داراست. انبار پیاده سازی هنگامیکه درصدد متصل شدن به شیئی خاصی است مورد استفاده قرار می گیرد یا هنگامی که می خواهد یک شیئی جدید را فعال کند مورد استفاده قرار میگیرد.

۲- Interface Definition Language- IDL

اگر یکی از پایه های ORB ، corba باشد، پایه دیگر آن IDL است IDL . زبانی است که به منظور تعریف واسط بین اجزاء نرم افزارها استفاده میشود IDL. یک زبان رویه ای نیست. آن فقط می تواند واسط را تعریف کند و پیاده سازی وظیفه آن نیست. برنامه نویسان ++C تعریف را مشابه سرفایلها و کلاسها فکر می کنند. در حقیقت IDL زبانی استاندارد برای تعریف واسط است. خصوصیات IDL ها از این مطمئن می سازد که داده ها بدون هیچ مشکلی بین دو زبان برنامه نویسی غیر مشابه ردو بدل شوند . استقلال از زبان IDL یک بخشی از استاندارد corba است و از هر زبانی مستقل است این استقلال نتیجه مفهوم نگاهت زبان است OMG . تعدادی از نگاهتهای زبان استاندارد را برای زبانهای محبوب تعریف کرده است.

۳- مدل شیئی corba هر معماری شیئی گرا یک مدل شیئی ای ارائه می دهد که چگونگی ارائه شدن اشیاء را در سیستم تشریح می کند corba . نیز یک سیستم شیئی گرا است در نتیجه دارای یک مدل شیئی است. بدلیل اینکه corba یک معماری توزیع شده است بهر حال ممکن است

مدل شیئی آن با مدل شیئی سیستمهای دیگر مانند ++C و java تفاوت داشته باشد. سه تفاوت عمده بین مدل شیئی corba و مدل شیئی سیستمهای تجاری دیگر وجود دارد که بشرح زیر است:

-حالت نیمه نمایش به منظور حمایت از توزیع اشیاء

-رفتار آن با ارجاعات اشیاء

-استفاده از سازگار دهنده اشیاء.

۴- Corba و مدل شبکه ای

حال باید جایگاه corba در مدل شبکه ای بررسی شود. نرم افزارهای corba روی لایه پروتکل IIOP نوشته می شوند. این پروتکلها خودشان نیز روی TCP/IP قرار دارند. نرم افزارهای تحت corba محدود نیستند که فقط از این پروتکلها استفاده کنند

۵- مشتریها و خدمتگذارها در corba

تعاریف خدمتگذار و مشتری در corba دقیقا مانند تعاریف اینها در سیستم مشتری - خدمتگذار است و میتوان اینگونه تشریح کرد که هر شیئی که یک متد را فراخوانی کند نقش مشتری را خواهد داشت. و هر شیئی که دارای متد در خواست شده باشد نقش خدمتگذار را خواهد داشت.

بیکر بندیها و ریشه ها Stubs and Skeletons

بعد از اینکه یک توسعه دهنده تعاریف واسط اجزاء را با IDL ایجاد کند فایل IDL خروجی را با کامپایلر IDL ایجاد می کند که با نام ریشه مشتری و استخوانبندی های خدمتگذار. اینها نقش یک چسب را بازی میکنند که مشخصات واسطهای مستقل از زبان را به یک کد پیاده سازی شده وابسته به زبان متصل می کند. استخوان بندی خدمتگذار قالب کاری را که خدمتگذار بر مبنای آن ت عریف شده است را فراهم می کند.

