

SPIN

درس سیستم عامل پیشرفته

علیرضا سلطانی نشان

۴ آبان ۱۴۰۲

فهرست مطالب

۱	۱ مقدمه
۱	۲ منظور از Dynamic loading
۲	۳ زبان برنامه نویسی Modula
۳	۴ برنامه نویسی دستوری یا Imperative Programming
۳	۵ موضوع Operator overloading

۱ مقدمه

سیستم عامل SPIN یک پروژه تحقیقاتی که با زبان برنامه نویسی ۳-Modula توسعه و پیاده‌سازی شده است، می‌باشد. این سیستم عامل یک پروژه متن باز است که برای سه هدف اصلی طراحی و پیاده‌سازی شده است:

۱. انعطاف پذیری^۱

۲. ایمن^۲

۳. اجرای مناسب^۳

این پروژه در دانشگاه واشنگتن^۴ توسعه داده شد.

در این سیستم عامل، کرنل می‌تواند توسط Dynamic loading ماژول‌ها که با اینترفیس‌ها پیاده‌سازی شده اند دامنه اجرایی را مشخص می‌کند. این دامنه‌ها توسط زبان برنامه نویسی مشخص شده‌اند. تمام اکستنشن‌های کرنل زیر مجموعه ایمن زبان ۳-Modula با ساختارهای

^۱Flexibility

^۲Safety

^۳Performance

^۴Washington of University

MetaLanguage و نوع تایپ سیف سیستم نوشته شده‌اند. همچنین این سیستم اکستنشن ران‌تایم کامپایلر را گزارش می‌کند. یک مجموعه‌ای از اکستنشن‌های کرنل یک Api را ارائه می‌دهند که می‌تواند سیستم دیجیتالی یونیکس را برای صدا زدن آن اینترفیس‌ها، شبیه سازی کند.

۲ منظور از Dynamic loading

موضوع Dynamic loading یک مکانیزمی است که توسط آن یک برنامه کامپیوتری می‌تواند در هنگام اجرا^۵ یک کتابخانه را چه به صورت یک منبع برنامه یا به صورت باینری، داخل حافظه رم خوانده، تمام آدرس‌های مربوط به توابع و متغیرهایی که داخل کتابخانه هستند را دریافت کرده، این توابع را اجرا کرده یا به متغیرها دسترسی داشته و در نهایت پس از انجام کارهایش کتابخانه را از داخل حافظه تخلیه می‌کند و اصطلاحاً حافظه را از آن کتابخانه آزاد می‌کند.

Dynamic loading به یک برنامه کامپیوتری اجازه اجرا در حالت نبود کتابخانه‌ها و وابستگی‌های مربوط به آن برنامه، می‌دهد که می‌تواند کتابخانه‌های موجود را پیدا کند، و به صورت بلقوه بتواند توابع و قابلیت‌های اضافی که در برنامه گنجانده نشده است را بدست آورد و کاری کند که نرم‌افزار بدون مشکل اجرا شود.

۳ زبان برنامه نویسی Modula

Modula-۳ یک زبان برنامه نویسی است که به عنوان جانشین آپگرید شده نسخه ۲-Modula شناخته می‌شود. این زبان در حالی که در محافل تحقیقاتی تاثیر گذار بوده است مانند حضور بین زبان‌های Python Java C# و حتی Nim اما هیچ وقت در حوزه صنعتی برای اهداف مختلف سازگار و مورد استفاده عموم قرار نگرفت.

ویژگی اصلی این زبان سادگی و ایمنی در حالی که تمام قدرت زبان برنامه نویسی-سیستمی را حفظ کرده است. این زبان با هدف ادامه سنت زبان پاسکال که به تایپ سیفتی معروف بود در حالی معرفی شد که سازه‌های جدیدی برای برنامه نویسی در دنیای واقعی را مطرح کرد. در عمل این زمان از موارد زیر به صورت کامل پشتیبانی می‌کند:

۱. برنامه نویسی Generic مشابه با Template. در حقیقت این پشتیبانی به برنامه نویس این قابلیت را ارائه می‌دهد که یک تایپ جنریک برای قسمت مورد نظر خود در نظر بگیرد. برای مثلا اگر به دنبال تعریف یک کلاس هستید می‌توانید داده‌های مربوط به

^۵time Run

کانستراکتور را از نوع جنریک تعریف کنید که کلاس شما بتواند در مرحله نمونه برداری به چند تایپ برای مورد خاص نقش ایفا کند.

۲. برنامه نویسی چند نخه یا Multithread Programming

۳. قابلیت جلوگیری و پیشبینی خطا یا Exception error handler

۴. قابلیت بسیار مهم Garbage collection

۵. برنامه نویسی از نوع شیء گرا یا Object Oriented Programming

هدف نهایی طراحی این زبان پیاده‌سازی ویژگی‌های مهم برنامه نویسی مدرن دستوری یا Imperative Programming در فرمی بسیار پایه و ابتدایی بوده است. بدین ترتیب به قول معروف در این زبان از پیاده‌سازی ویژگی‌های پیچیده و خطرناک از قبیل چند وراثتی و Operator overloading چشم پوشی شده است.

۴ برنامه نویسی دستوری یا Imperative Programming

در حقیقت در این نوع از برنامه نویسی، تمرکز بر تعریف آن که چگونه یک برنامه به صورت مرحله به مرحله عمل می‌کند می‌باشد بجای آن که بیشتر روی توصیف انتظار روی نتیجه آن در سطوح بالاتر وجود داشته باشد.

همانطور که در این شیوه از برنامه نویسی مشخص است، تمام افعال به صورت دستوری خواهد بود. برای مثال زمانی که می‌خواهیم داده‌ای را از یک جدولی از پایگاه داده دریافت کنیم به صورت کلی می‌گوییم که داده مربوط به دانشجوی با شناسه ۱۲ را دریافت کن. در حقیقت در این شیوه ما نتیجه برنامه را بیان کردیم. اما در شیوه دستوری به اینگونه عمل نمی‌کنیم. در حقیقت تمام مراحل را مانند الگوریتم مرحله به مرحله توضیح می‌دهیم. ابتدا به پایگاه داده متصل شو، یک کوئری بنویس که حاوی شرط‌های مورد نظر باشد، داده‌ها را دریافت کن. بعد از دریافت داده‌ها آنها را نرمال‌سازی کن. سپس داده‌ها را به سمت جریان (ب) ارسال کن. اگر ارسال داده موفقیت آمیز بود ارتباط با پایگاه داده را ببند. وضعیت را موفقیت آمیز اعلام کن.

۵ موضوع Operator overloading

یکی از موارد مشخص برنامه نویسی شیء گرا و استفاده از قانون چندریختی یا Polymorphism می‌باشد. این بخش به طور کامل در سندی داخل این مخزن توضیح داده خواهد شد.