

تفاوت کامپایلر و مفسر

نیما خدادادی



<https://blog.faradars.org>

<https://developersho.com>

<https://7learn.com>

-1در زبان‌های کامپایلری قبل از اجرای برنامه، همه‌ی کدها به یکباره به «زبان ماشین» تبدیل می‌شوند. اما در زبان‌های مفسری موقع اجرای برنامه، کدها به‌صورت خط به خط به زبان ماشین ترجمه می‌شوند.

-2در سرعت اجرا بدون شک، زبان‌‌های کامپایلری در سرعت اجرای برنامه‌ها بهتر هستند. چرا که کامپایلر، کد را به صورت کامل به زبان ماشین ترجمه کرده و خروجی آن بهینه و سریع است. در مقابل، زبان‌‌های مفسری برای هر دستوری که باید اجرا شود، باید کدها را تفسیر و سپس آن را اجرا کنند که باعث کاهش سرعت اجرای برنامه می‌شود.

-3اگر خطایی در کد زبان‌های برنامه نویسی کامپایلری رخ بدهد، کامپایلر با خطای Compile error مواجه می‌شود و فایل خروجی ساخته نمی‌شود و نمی‌توان آن را اجرا کرد. اگر در کد زبان‌های برنامه نویسی مفسری خطایی رخ دهد، چون به‌صورت خط به خط این ترجمه انجام می‌شود، برنامه تا زمانی که خطایی در کد وجود نداشته باشد اجرا شده و موقع رسیدن به آن خط از کد که خطا وجود دارد، برنامه متوقف می‌شود.

-4سرعت اجرای برنامه در زبان‌های کامپایلری به دلیل اینکه خروجی کامپایل شده برای دفعات بعدی نیاز به ترجمه نیس، بالاتر از سرعت اجرای برنامه در زبان‌های مفسری است چون که همیشه موقع اجرا این ترجمه باید انجام شود و مقداری زمان‌بر است.

-5پرتاب‌های استثناء (exception handling): برخلاف زبان‌‌های کامپایلری، زبان‌‌های مفسری به صورت پیش فرض پرتاب استثناء دارند. این به معنای این است که در صورت بروز خطا در هنگام اجرای برنامه، برنامه متوقف و خطایی به کاربر نمایش داده می‌شود.

-6زبان مفسری وابسته به سیستم‌عامل نیست اما زبان کامپایلری به سیستم‌عامل وابسته است. در زبان مفسری کدها چون در هر اجرای برنامه ترجمه می‌شوند، در هر سیستم‌عاملی قابل اجرا هستند. اما خروجی زبان کامپایلری، در همان سیستم‌عاملی قابل اجرا است که در همانجا کامپایل شده است. اگر بخواهیم در سیستم‌عامل دیگری استفاده کنیم، باید در همان سیستم‌عامل، کدها دوباره کامپایل شوند.

7- در رویکرد مفسرها هیچ کد ماشینی وجود ندارد؛ زیرا این برنامه‌ها به صورت خط به خط ترجمه و تفسیر می‌شوند. اما در روش استفاده از کامپایلر، کدهای ماشین با زبان ماشین روی حافظه دیسک ذخیره خواهند شد

8- کامپایلر بر اساس روش‌های Linking and Loading ترجمه برنامه نوشته شده را انجام می‌دهد. بارگذاری و اتصال در برنامه‌ها، نقش مهمی را برای پیاده‌سازی برنامه انجام می‌دهند. روش اتصال، کدهای شی تولید شده توسط اسمبلر را دریافت و آن‌ها را برای تولید ماژول پیاده‌سازی ترکیب می‌کند. در سمتی دیگر، روش بارگذاری، ماژول اجرایی را برای پیاده‌سازی در حافظه اصلی بارگذاری کرده است. اما مفسر بر اساس Method تفسیر را انجام می‌دهد و با استفاده از آن هر زبان برنامه نویسی با متد مخصوص به خود خط‌های برنامه را تفسیر خواهد کرد.

9- برنامه‌های کامپایلری به طور مستقیم و یک‌جا پیاده‌سازی می‌شوند و در رویکرد آن‌ها نیازی به ذخیره برنامه نوشته شده کامپایلر در حافظه سیستم وجود ندارد. اما، در برنامه‌ای که با مفسر ترجمه می‌شود، هنگام تفسیر آن به صورت خط به خط، از حافظه سیستم استفاده خواهد شد.

10- به طور کلی می‌توان گفت که کامپایلر‌ها زمان نسبتاً بیشتری برای کامپایل کردن کدهای کوچک صرف می‌کنند که ممکن است چندین بار اجرا شوند. به همین دلیل برای پیاده‌سازی کدهای کوچک و کوتاه استفاده از برنامه‌های مفسری بهتر است.

11- برنامه نویسی با رویکرد «تعیین نوع به صورت پویا» یا «نوع‌دهی پویا» (Dynamic Typing) برای زبان‌های برنامه نویسی کامپایلری دشوار است، زیرا آن‌ها نمی‌توانند پیش‌بینی کنند که در هر بخش از پیاده‌سازی چه اتفاقی رخ خواهد داد. اما زبان‌‌های مفسری از قابلیت تعیین نوع به صورت پویا به طور کامل پشتیبانی می‌کنند.

12-کامپایلر به دلیل اینکه سرعت بالاتری دارد، برای محیط تولید محصول مناسب‌تر است. مفسرها برای برنامه‌ها و محیط‌های توسعه مناسب هستند؛ زیرا در این محیط‌ها بررسی خط به خط کدها اهمیت دارد و به سرعت پیاده‌سازی کمتر توجه می‌شود. همچنین می‌توان از مفسر در پروژه‌هایی استفاده کرد که در آن‌ها زمان بارگذاری کدها مهم است، برای مثال می‌توان به محیط‌های توسعه وب اشاره کرد.