



کامپایلرهای AOT , JIT در دات نت کور

رحیم قاسمی هتم یک برنامه نویس و معمار سیستم مینمال هتم

- در دات نت کور، دورویکند اصلی برای کامپایل کردن برنامه ها وجود دارد:

- کامپایلر Just-In-Time (JIT)

- و کامپایلر Ahead-Of-Time (AOT).

- هر دوروش کاربردهای متفاوتی دارند و مزایا و معایب خاص خود را ارائه می دهند. در

ادامه به بررسی هر کدام می پردازیم:

- کامپایلر: Just-In-Time (JIT)

- نحوه عملکرد: کامپایلر JIT کد بایت (IL زبان میانی) که توسط کامپایلر C# تولید می شود را در زمان اجرا به کد ماشین تبدیل می کند. این به این معنی است که کد برای معماری ماشین که روی آن اجرا می شود کامپایل می شود.

- مزایا:

- زمان راه اندازی سریعتر: از آنجایی که کامپایل در زمان اجرا اتفاق می افتد، برنامه ها به سرعت راه اندازی می شوند.

- بهینه سازی پویا: کامپایلر JIT می تواند به طور مداوم کد در حال اجرا را تجزیه و تحلیل کند و آن را بر اساس الگوهای استفاده برای عملکرد بهینه کند.

- بهره وری حافظه: کد فقط در صورت نیاز کامپایل می شود، که باعث کاهش فضای اشغال شده در حافظه می شود.

• معایب JAL:

- اجرای اولیه کندتر: اجرای اولیه می‌تواند کندتر باشد زیرا که باید ابتدا کامپایل شود.
- استفاده بیشتر از حافظه در حین کامپایل: برای فرآیند کامپایل به حافظه اضافی نیاز است.
- پشتیبانی محدود از پلتفرم: برنامه‌های کامپایل شده با JAL به وجود **NET Runtime**. در سیستم نیاز دارند.

- کامپایلر: Ahead-Of-Time (AOT)

- نحوه عملکرد: کامپایلر AOT کد بایت IL را قبل از استقرار برنامه به کد ماشین تبدیل می‌کند. این کار منجر به فایل اجرایی متقل می‌شود که می‌تواند بدون نیاز به .NET Runtime. به طور متقیم بر روی سیستم هدف اجرا شود.

- مزایا:

- زمان اجرای سریعتر: کد از قبل کامپایل شده است، که نیاز به کامپایل در زمان اجرا را از بین می‌برد.
- فضای اشغال شده در حافظه کمتر: .NET Runtime مورد نیاز نیست، که منجر به اندازه کوچکتر برنامه می‌شود.

- عملکرد بهتر: کد Native می‌تواند often faster than JIT-compiled code اجرا شود.

- پشتیبانی وسیع تر از پلتفرم: برنامه‌های کامپایل شده با AOT می‌توانند بر روی سیستم‌هایی بدون .NET Runtime اجرا شوند.

- معایب:

- چرخه توسعه کندتر: کامپایل AOT باید برای هر پلتفرم هدف انجام شود، که زمان توسعه را

افزایش می دهد.

- بهینه سازی پویای محدود: کد کامپایل شده با AOT به طور پویا مانند کد کامپایل شده با JIT قابل

optimize نیست.

- اندازه بزرگتر برنامه: کد Native کامپایل شده برای هر پلتفرم باید در برنامه گنجانده شود.

- انتخاب بین AOT و JIT

- انتخاب بین کامپایل AOT و JIT به نیازهای خاص برنامه شما بستگی دارد. در اینجا چند فاکتور وجود دارد که باید در نظر بگیرید:

- **نوع برنامه:** AOT برای برنامه‌های به طور مداوم برای مدت طولانی اجرا می‌شوند و عملکرد برای آنها بسیار مهم است به عملکرد مانند خدمات ابری یا برنامه‌های سمت سرور مناسب‌تر است. ممکن است برای برنامه‌های دسکتاپ یا موبایل که زمان راه اندازی در آنها اهمیت کمتری دارد، کافی باشد.

- **پلتفرم استقرار:** اگر برنامه را برای پلتفرم‌هایی که **NET Runtime** به طور **readily available** نیست منتشر می‌کنید، AOT تنها گزینه است.

- **چرخه توسعه و استقرار:** اگر به تعخیرات و **deployments** مکرر نیاز دارید، JIT به دلیل چرخه توسعه سریعتر ممکن است ترجیح داده شود.

*در نتیجه،

هر دو کامپایل AOT و JIT نقاط قوت و ضعف خود را دارند.

درک این تفاوت ها و در نظر گرفتن نیازهای خاص برنامه شما به شما کمک می کند تا مناسب ترین رویکرد را برای عملکرد بهتر انتخاب کنید .



نظراتتون رو با من به اشتراک بذارید .

برای شرکت در کلاس ها و اطلاعات بیشتر

با من در ارتباط باشید .

۰۹۹۲۲۹۳۷۵۶۵