عنوان: **اعمال غیر همزمان و چند ریسمانی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۳ ۲۳:۳۷
آدرس: www.dotnettips.info
گروه‌ها: C#, Asynchronous Programming

تصور عموم بر آن است که اعمال غیر همزمان با چند ریسمانی به یک معنا هستند. این مورد الزاما صحیح نیست. برای مثال دریافت غیرهمزمان یک فایل را از اینترنت درنظر بگیرید. شاید اینطور به نظر برسد که در اینجا یک ترد جدید ایجاد شده و در آن کل کار دریافت فایل آغاز می‌گردد؛ اما خیر. ایجاد یک ترد جدید تنها در قسمت‌های خاصی از یک پروسه انجام می‌شود. همچنین از لحاظ فنی امکان انجام کل کار در یک ترد، بدون بلاک کردن آن وجود دارد. از این جهت که بیشتر زمان، جهت صبر کردن دریافت پاسخی از سرور صرف می‌شود. زمانیکه کلاینت درخواستی را ارسال می‌کند، دیگر کار خاصی را نمی‌تواند انجام دهد تا اینکه پاسخی را دریافت کند.

زمانیکه از یک API غیرهمزمان برای مدیریت چنین عملیاتی استفاده می‌شود، ترد جاری را در این حالت در خواب فرو می‌برد. برای اینکه کار بیشتری برای انجام وجود ندارد. همچنین با اینکه کلاینت درخواستی را ارسال می‌کند یا پاسخی را دریافت، برای مدیریت کل عملیات در اکثر اوقات نیازی به تردها ندارد. این سخت افزار شبکه‌ی نصب شده در سیستم است که عمده‌ی کار را انجام می‌دهد و نه برنامه. زمانیکه برنامه درخواست ارسال اطلاعاتی را بر روی شبکه ارائه می‌دهد، درایور سخت افزار شبکه است که به سخت افزار مرتبط فرمان می‌دهد چه اطلاعاتی را باید ارسال کند. اکثر اینگونه سخت افزارها قادرند اطلاعات را خارج از حافظه‌ی اصلی سیستم دریافت کنند. در اینجا درایور تنها باید به سخت افزار عنوان کند، چه اطلاعاتی را و به کجا باید ارسال کند. بنابراین CPU تنها در طی ارسال این فرمان است که مشغول می‌باشد و نه خارج از آن و این زمان اصلا در مقایسه با زمان ارسال اطلاعات توسط سخت افزار مرتبط، طولانی نیست. CPU مجددا زمانی درگیر خواهد شد که سخت افزار شبکه، اطلاعاتی را دریافت کرده است و باز هم این زمان در مقایسه با زمان دریافت اطلاعات توسط سخت افزار شبکه بسیار کوتاه است.

اغلب کارهای IO به همین شکل هستند. شبیه به همین روند در حالت دسترسی به سخت دیسک وجود دارد. مدت زمانیکه CPU به دیسک کنترلر اعلام می‌کند چه اطلاعاتی را نیاز دارد در مقایسه با مدت زمانیکه دیسک کنترلر این اطلاعات را واقعا بارگذاری می‌کند، بسیار ناچیز است.

نمونه‌ی دیگر آن کار با بانک‌های اطلاعاتی است. در اغلب اوقات برنامه‌ی ما صرفا یک درخواست را به بانک اطلاعاتی ارائه می‌دهد و اصل عملیات در جایی دیگر و توسط موتور بانک اطلاعاتی، خارج از برنامه پردازش می‌گردد.

بنابراین جهت پردازش یک پروسه‌ی خاص، در بسیاری از مراحل آن تنها یک ترد کافی است و هدف اصلی اعمال غیرهمزمان، کاهش تعداد تردهایی است که برنامه جهت پردازش عملیاتی خاص، نیاز دارد. این نوع الگوریتم‌ها طوری طراحی شده‌اند تا تردها تنها زمانی بکار گرفته شود که واقعا CPU قرار است کار خاصی را انجام دهد و نه برای مثال زمانیکه دیسک کنترلر یا سخت افزار شبکه مشغول به کار هستند (و ویندوز به صورت توکار دارای یک چنین API ایی هست). این مساله در سمت کلاینت، سبب خواهد شد تا ترد UI آزاد شود و بتواند به درخواست‌های رسیده کاربر بهتر پاسخ دهد. همچنین این مساله در سمت سرور نیز بسیار مفید است، زیرا برنامه قادر خواهد شد تا به تعداد بیشتری از درخواست‌ها به صورت همزمان پاسخ دهد. زیرا با کاهش تعداد تردهای درگیر، مقیاس پذیری سیستم افزایش می‌یابد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۳ ۱۱:۴۱

یک مطلب تکمیلی

توضیحات بیشتری در مورد اینکه پردازش اعمال غیرهمزمان واقعی پشتیبانی شده توسط ویندوز، نیازی به ترد اضافی ندارند:

There Is No Thread