

بسمه تعالی



تمرین شماره ۳
سیستم‌های عامل

محسن کربلایی امینی، ۹۸۲۴۲۱۲۸

آبان ۱۴۰۲

سوال ۱:

مثال: جمع کردن خانه‌های یک آرایه با هم.

- **Data Parallelism**: اجرای همروند (*concurrent*) یک تسک روی هسته‌های پردازشی. در مثال بالا در صورت داشتن دو هسته، یک رشته (*thread*) می‌تواند پردازش مربوط به $n/2$ عنصر اول را انجام دهد و هسته‌ی بعدی می‌تواند عناصر $n/2$ تا n را باهم جمع کند.
- **Task Parallelism**: اجرای همروند تسک‌های مختلف روی هسته‌های پردازشی. رشته‌های مثال بالا را نمی‌توان روی هسته‌های مختلف برد. البته بر روی هسته‌های مختلف، رشته‌ها در حال استفاده از منابع پردازشی هستند، اما هر رشته مربوط به تسکی متفاوت خواهد بود.

سوال ۲:

این مفهوم اشاره به این دارد که ساختاری که یک رشته نیاز دارد از قبل ساخته شده و آماده به کار هست. یعنی به محض اینکه برای یک رشته کاری تعریف شود از *thread pool* خارج و شروع به انجام آن کار می‌کند. برای این مفهوم می‌توان سرویس‌های آژانس قدیم را مثال زد، همواره تعدادی راننده آماده به ارائه سرویس بودند و به محض درخواست مشتری یکی از راننده برای مشتری فرستاده می‌شود.

سوال ۳:

- توان عملیاتی (*throughput*) بالاتر به دلیل امکان همروندی محاسباتی پردازشی و *I/O*
- پاسخ‌دهی بهتر برنامه‌ها، اگر یک درخواست روی رشته مربوط به خود اجرا شود، برنامه‌ها *freeze* و بلاک نمی‌شوند تا جواب آن رشته برگردد.
- مصرف بهتر منابع سیستم به این دلیل که رشته‌ها سربار کمتری نسبت به ایجاد پراسس‌ها دارند.
- پهنای‌بند بیشتر، و تاخیر پایین‌تر در به اشتراک‌گذاری حجم‌های زیادی از دیتا در استفاده از رشته‌های مختلف در یک فضای آدرس حافظه‌ی مشترک.

سوال ۴:

User thread ها از سمت کرنل قابل مشاهده و دسترسی نیستند. یک برنامه می‌تواند چندین *user thread* داشته باشد، اما اگر یکی از این *thread* ها بلاک شود، کل آن برنامه بلاک خواهد شد. به این ترتیب اگر ارتباط

بین *user thread* ها برقرار باشد، این امکان وجود دارد که در صورت بلاک شدن یکی از رشته‌های برنامه، فقط آن رشته بلاک شود و اجرای بقیه رشته‌های برنامه متوقف نشود.
روش‌های ایجاد این ارتباط عبارتند از:

- *Many-to-One*
- *One-to-One*
- *Many-to-Many*

سوال ۵:

برای محاسبه بهبود این بهبود سرعت می‌توانیم از قانون آمثال استفاده کنیم:

$$speedup \leq \frac{1}{S + \frac{(1-S)}{N}}$$

$$speedup \leq \frac{1}{0.4 + \frac{0.6}{2}} = \frac{1}{0.7} = \frac{10}{7} = 1.42$$

در این انتقال سرعت ۴۲ درصد بهبود سرعت خواهیم داشت.
با افزایش تعداد پردازنده‌ها به سمت بی‌نهایت، یعنی حد عبارت بالا هنگامی که N به سمت بی‌نهایت برود، عبارت $\frac{1}{S}$ را خواهیم داشت. این بدین معنی است که با افزایش تعداد هسته‌ها، در واقع بخش موازی پردازش، گویی در زمان صفر توسط بی‌نهایت هسته پردازش می‌شود و فقط بخش سریال پردازش باقی می‌ماند.

سوال ۶:

سیستم‌هایی که در قالب وب‌سرور ارائه می‌شوند را می‌توان مثالی از این دست دانست. این سیستم‌ها در صورت تکرشته‌ای بودن، هر درخواست یک *client* را به صورت *blocking* پاسخ می‌دهند. به این معنی که تا درخواست کاربر قبلی پردازش و پاسخ داده نشده باشد، درخواست کاربر بعدی باید منتظر بماند. در نتیجه انتقال به معماری *multi-threaded* برای این سیستم‌ها ضروری می‌باشد.

سوال ۷:

توضیحات	مقدار a
اجرای پشت سرهم T^1 و T^2 و یا اینکه یک خط از T^2 و سپس یک خط از T^1 اجرا شود	۶۰

	اجرای خط اول از T^2 ، خط اول از T^1 ، ادامه و اتمام T^2 ، ادامه و اتمام T^1
.	اجرای پشت سر هم T^1 و T^2 و یا اینکه یک خط از T^1 و سپس یک خط از T^2 اجرا شود

در هر صورت در این برنامه، شرط داخل T^1 درست خواهد بود و هیچوقت z مقداری زیر صفر نخواهد داشت.

سوال ۸:

الف) درست. پردازش موازی برای سیستم‌هایی امکان‌پذیر است که چند پردازنده داشته باشند اما همروندی برای همه سیستم‌ها ممکن است.

ب) نادرست. پاسخ کامل این نیاز در پاسخ سوال ۴ داده شده است. البته نیاز به معنی الزام وجود ندارد اما نیاز به دلیل بهبود کارایی سیستم وجود دارد.

ج) نادرست. یک پراسس می‌تواند چند رشته *One-to-One* داشته باشد تا از این مشکل جلوگیری کند. البته تعداد این رشته‌ها بعضاً محدود شده‌است.