

پاییز ۱۳۹۶

سیستم‌های توزیعی

تمرین دوم

پرهام الوانی

۹۶۱۳۱۱۱۲

سوال ۱

(الف) سه مورد از آنچه باعث ایجاد تفاوت میان Local Procedure Call و Remote Procedure Call می‌گردند عبارتند از:

۱. دسترسی به حافظه

۲. تاخیر

۳. شکست‌های جزئی^۱

(ب) در مورد دسترسی به حافظه روش‌های مختلفی را می‌توان استفاده کرد، یکی از این روش‌ها copy/restore است، در این روش یک کپی از قسمتی از حافظه به صورت reference می‌باشد به سمت سرور فرستاده می‌شود و پاسخ سرور در آن حافظه بازنویسی می‌گردد. البته این روش تمام آنچه برای دسترسی به حافظه ممکن است در یک برنامه رخ دهد را نمی‌تواند پشتیبانی کند.

در مورد شکست‌های جزئی راه حل‌های مختلفی وجود دارد که ساده‌ترین آن‌ها روش strawman است که در آن شکست‌های جزئی سیستم را به شکست کل سیستم تبدیل کنیم (!). راه حل واقعی از بین بردن transparency و پشتیبانی از یکی از semanticهای exactly-once, at-least-once و ... می‌باشد. مشکل تاخیر هم به نوعی در همین بخش می‌گنجد، زیرا بزرگترین سختی تشخیص تاخیر از شکست است و برای حل موضوع نیز مجبور هستیم transparency را از بین برده و از یکی از همان semanticهای معرفی شده استفاده کنیم.

سوال ۲

(الف) ذخیره‌سازی یک فایل با توجه به ساختار فایل سیستم توزیع شده می‌تواند حالت‌های مختلفی داشته باشد، اگر تکرار نوشتن یک فایل صرفاً منجر به بازنویسی آن گردد می‌توان از at-least-once استفاده کرد، این در حالی است که اگر این تکرار باعث خطا شود می‌بایست از at-most-once استفاده کرد.

(ب) خرید یک لپ‌تاپ به صورت آنلاین را نمی‌توان تکرار کرد بنابراین می‌بایست از at-most-once استفاده کرد.

(ج) این عمل را می‌توان بدون هیچ مشکلی تکرار کرد، پس استفاده از at-least-once منطقی به نظر می‌رسد.

(د) آپلود یک پست در وبلاگ اگر تکرار شود باعث تکرار پست می‌گردد بنابراین می‌بایست از at-most-once استفاده کرد.

سوال ۳

(الف)

عدالت: در راه حل پیشنهادی نباید هیچ فیلسوفی گرسنه بماند، یعنی بعد یک بازه متناهی می‌بایست هر فیلسوف یکبار نوبت غذا خوردن داشته باشد.

¹ Partial Failures

کارایی: راه حل پیشنهادی باید پیشرفت داشته باشد، به این معنا که نباید در آن Deadlock رخ دهد. در این مساله یک مثال Deadlock به این صورت است که هر فیلسوف یک چنگال داشته باشد و منتظر چنگال نفر کناری خود باشد
صحت: در روش پیشنهادی نباید حالتی وجود داشته باشد که دو فیلسوف همزمان برای یک چنگال تقاضا دهند.

(ب)

در این روش ممکن است Deadlock رخ دهد به این صورت که اگر هر فیلسوف چنگال سمت چپ خود را برداشته باشد وارد وضعیتی می‌شود که می‌بایست چنگال سمت راست را بردارد که این در حالی است که چنگال سمت راست در دست فیلسوف دیگری است که او نیز در انتظار نفر کناری خود است و به همین ترتیب ...

(ج)

در این روش می‌بایست یک الگوریتم برای بلند شدن فیلسوف‌ها از سرمیز قرار گرفت، اگر چنین الگوریتمی در نظر گرفته نشود، عدالت از دست می‌رود زیرا ممکن است هرگز کسی که سر میز ننشسته است فرصت این را پیدا نکند که سر میز بنشیند.