



دانشگاه علم و صنعت ایران

دانشکده مهندسی کامپیوتر

دکتر رضا انتظاری ملکی

پاییز ۱۴۰۰

تمرین تئوری سری دوم

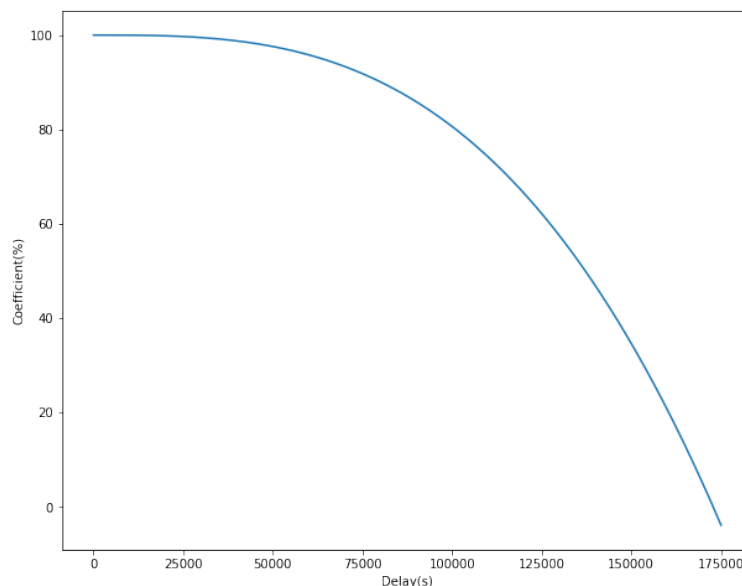
سیستم‌های عامل

دانیال بازمانده - محمدعلی فراهت

تاریخ تحویل: ۲ آبان ساعت ۲۳:۵۹:۵۹

قوانین

- در صورت مشاهده هرگونه تقلب، به ازای هربار تقلب نمره‌ی کل آن تمرین صفر در نظر گرفته می‌شود و همچنین یک نمره (نمره منفی) از نمره‌ی کل تمرین‌ها کسر می‌شود.
- در صورت وجود هرگونه سوال از طریق گروه تلگرام مطرح کنید. (لطفا پی‌وی پیام ندهید).
- ۱۰ درصد از نمره‌ی هر تمرین به تمیزی و نظم پاسخ‌های ارسالی شما تعلق گرفته است. لازم است به موارد زیر توجه کنید:
 ۱. خوانا و مرتب بنویسید.
 ۲. از نرم‌افزارهای مناسب جهت اسکن کردن تمرین‌های خود استفاده کنید و چک کنید که نور تصاویر مناسب هستند. مانند: CamScanner, Microsoft Office Lens, Adobe Scan, ...
 ۳. به طور عمودی عکاسی کنید.
 ۴. پاسخ هر سوال را به طور جداگانه در کوئرا آپلود کنید.
- محور افقی این نمودار، مقدار تاخیر به ثانیه و محور عمودی ضریب اعمالی در نمره تمرین است.



شکل ۱: نمودار تاخیر

سوالات

۱ User-level vs Kernel-level (۱۵ نمره)

دو تفاوت اصلی user-level threads و kernel-level threads را نام ببرید. هریک تحت چه شرایطی از دیگری بهتر است؟

۲ Multicore Programming (۲۰ نمره)

آیا یک راه حل multithreaded می‌تواند با استفاده از چند user-level thread در سیستم‌های چندپردازنده‌ای، عملکرد بهتری نسبت به سیستم‌های تک‌پردازنده‌ای ایجاد کند؟ توضیح دهید.

۳ Concurrency vs Parallelism (۲۰ نمره)

پردازش موازی (Parallel) و همروند (Concurrent) را با رسم نمودار روند اجرا با یکدیگر مقایسه کنید.

۴ Amdahl's Law (۱۵ نمره)

ذهن خلاق و بهینه‌دوست: شما که بسیار به بازی‌های کامپیوتری علاقه دارید و از محدودیت‌های پردازشی موجود نیز باخبر است، به تازگی با مفهوم پردازش موازی آشنا شده و از شما می‌پرسد چرا به جای تعداد محدود هسته‌های محاسباتی موجود در پردازنده‌های امروزی، از تعداد بیشتری هسته (مثلاً ۱۰۰ هسته!) استفاده نمی‌شود تا از سرعت‌های چند برابر بهره بگیریم؟

پاسخ شما چیست؟ (نرخ افزایش سرعت را برای ۲، ۴، ۸ و ۱۰۰ هسته محاسبه کنید.)

۵ Multithreading Models (۲۰ نمره)

یک سیستم multicore و یک برنامه multithreaded را که با استفاده از مدل many-to-many threading نوشته شده است، در نظر بگیرید. فرض کنید تعداد user-level thread ها در برنامه بیشتر از تعداد هسته‌های محاسباتی در سیستم باشد.

در هر یک از حالت‌های زیر Performance را مورد بررسی قرار دهید.

- (آ) تعداد kernel thread های اختصاص داده شده به برنامه از تعداد هسته‌های پردازشی کمتر باشد.
- (ب) تعداد kernel thread های اختصاص داده شده به برنامه با تعداد هسته‌های پردازشی برابر باشد.
- (ج) تعداد kernel thread های اختصاص داده شده به برنامه از تعداد هسته‌های پردازشی بیشتر باشد اما از تعداد user-level thread ها کمتر باشد.

موفق باشید.