



تمرین کامپیوتری سوم

سیستم‌های عامل - پاییز ۱۳۹۹

گزارش کار

دانشکده مهندسی برق و کامپیوتر

نام و نام خانوادگی: امیرحسین عباسکوهی

تاریخ: 1399/9/24

استاد:

810197539

دکتر مهدی کارگهی

2	مقدمه
3	پیاده‌سازی سری
3	سوال اول
3	سوال دوم
3	جدول اول
3	پیاده‌سازی چندریسه‌ای
3	سوال سوم
4	سوال چهارم
4	سوال پنجم
4	جدول دوم

مقدمه



در این تمرین به تحلیل داده‌هایی که از مشخصات و قیمت فروش گوشی‌های موبایل جمع‌آوری شده است

پردازش شده است. در ابتدا برنامه اقدام به خواندن و تجزیه مجموعه داده¹ی ارائه شده می‌کند و آنها را در

حافظه خود ذخیره می‌کند. پس از استخراج داده‌ها و ویژگی‌های آنها، برنامه اقدام به نرمال‌سازی² داده‌ها و در نهایت اقدام به تعیین

طبقه قیمتی گوشی‌ها می‌کند. این تمرین به دو روش این مسئله پیاده‌سازی شده است که در ادامه گزارش، نتایج حاصل آمده است.



¹ Dataset

² Data Normalization

پیاده‌سازی سری

سوال اول

چرا برای پیاده‌سازی یک برنامه بصورت چندریشه‌ای، بهتر است ابتدا این برنامه بصورت سری پیاده‌سازی شود؟

این موضوع می‌تواند دلایل مختلفی داشته باشد. یکی از مهمترین دلایل ها این است که باید متوجه شویم کدام بخش های برنامه زمان بیشتری می‌طلبند و به اصطلاح HOTSPOT هستند تا بتوانیم آن نقاط را موازی سازی کنیم. دلیل دیگر پیچیدگی برنامه نویسی موازی است که تبدیل برنامه سری به موازی ساده تر از تبدیل این است که از اول برنامه را موازی بنویسیم.

سوال دوم

با بررسی زمان اجرای بخش‌های مختلف برنامه، $topstoH^3$ های برنامه را مشخص کنید.

با بررسی انجام شده بخش هایی که زمان زیادی از برنامه میگیرند: 1- خواندن کل فایل، 2- محاسبه max و min و normalize کردن داده ها 3- محاسبه دقت. این بخش چون باید کل برنامه را در یک مرحله حساب کنند وقت گیر هستند اما با موازی سازی میتوان برای مثال با 4 هسته این بخش ها را حساب کرد. در نتیجه زمان بهتری خواهیم داشت.

جدول اول

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را بازای ورودی نمونه‌ای که در شرح تمرین آمده است، در جدول زیر بیاورید.

³ توابعی که در برنامه‌تان بیشترین زمان اجراها را به خود اختصاص می‌دهند.

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اول
0.084	0.097	0.073	0.110	0.069	0.106	0.051

پیاده‌سازی چندریسه‌ای

سوال سوم

اگر هنگام موازی‌سازی برنامه به زمان اجرای بیشتری نسبت به حالت سری برخورد کنید، چه رویکردهایی را برای کاهش زمان اجرا و استفاده حداکثری از موازی‌سازی پیش می‌گیرید؟

البته من به این مشکل برخورد نکردم، اما از راه حل‌ها برای حل این موضوع تعداد مناسب thread ها می‌باشد. هم چنین تعیین دقیق نقاط Hotspot می‌باشد. اگر در جایی از کد از منابع استفاده شده (چون منابع مشترک اند)، در صورت افزایش ریسسه‌ها در این نقاط زمان افزایش می‌یابد. از موارد مهم دیگر در زمان mutex lock است که کاری کنیم در حالت کمتری به این موارد برخورد کنیم.

سوال چهارم

در هنگام پیاده‌سازی این بخش، به چه چالش‌هایی برخورد کردید و بیان کنید که به چه صورت آن‌ها را رفع کردید.

از موارد چالش برانگیز بحث lock متغیر برای عدم تداخل ریسسه‌ها بود که باید از mutex lock استفاده میشد. مورد دیگر هم که بسیار مهم بود، صبر کردن برای اتمام ریسسه‌ها پس از انجام کاری بود که باید با pthread_join انجام می‌شد زیرا مراحل به هم وابسته بودند و بدون اتمام یک مرحله شروع کردن مرحله بعد اشتباه بود.

سوال پنجم

با توجه به تجربه‌ای که در پیاده‌سازی این تمرین بدست آوردید، به نظر شما در چه مواقعی از قفل⁴ در یک طراحی چندریسه‌ای ضروری است؟ تاثیر استفاده از قفل‌ها را بر روی کارایی⁵ سامانه بیان کنید.

در مواردی که قرار است ریسه‌ها مقداری را تغییر دهند که بین ریسه‌ها مشترک است ممکن این ریسه‌ها در وسط کد اجرایی ماشین متوقف شوند در نتیجه عملیات تغییر آن مقدار کامل انجام نشود در مقدار متغیر قابل پیش بینی نیست و به احتمال بسیار بالایی مقدار اشتباه در نتیجه سیستم درست کار نمیکند و کارایی پایین می‌آید.

جدول دوم

زمان‌های اجرای ۶ اجرای متوالی از برنامه و میانگین آن‌ها را بازای ورودی نمونه‌ای که در شرح تمرین آمده است، در جدول زیر بیاورید.

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اول
0.073	0.083	0.071	0.088	0.050	0.100	0.048

میزان تسریع ($\frac{Serial\ Time}{Parallel\ Time}$) برنامه نسبت به حالت سری را در زیر بیاورید.

میزان تسریع	میانگین زمان اجرای موازی	میانگین زمان اجرای سری
1.15	0.073	0.084

⁴ Lock

⁵ Performance

