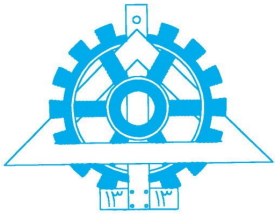
تمرین کامپیوتری سوم

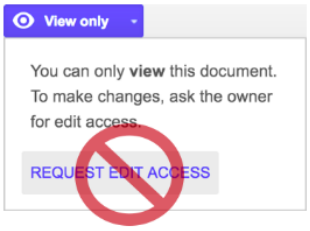


دانشکده مهندسی برق و کامپیوتر | گزارش کار | سیستم‌های عامل - پاییز ۱۳۹۹

استاد:
دکتر مهدی کارگهی

تاریخ:

نام و نام خانوادگی:

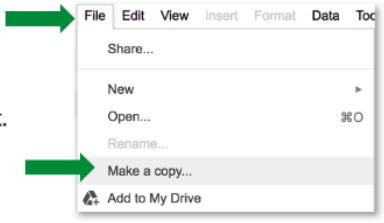


How to use this template:

This is a view-only file and cannot be edited.

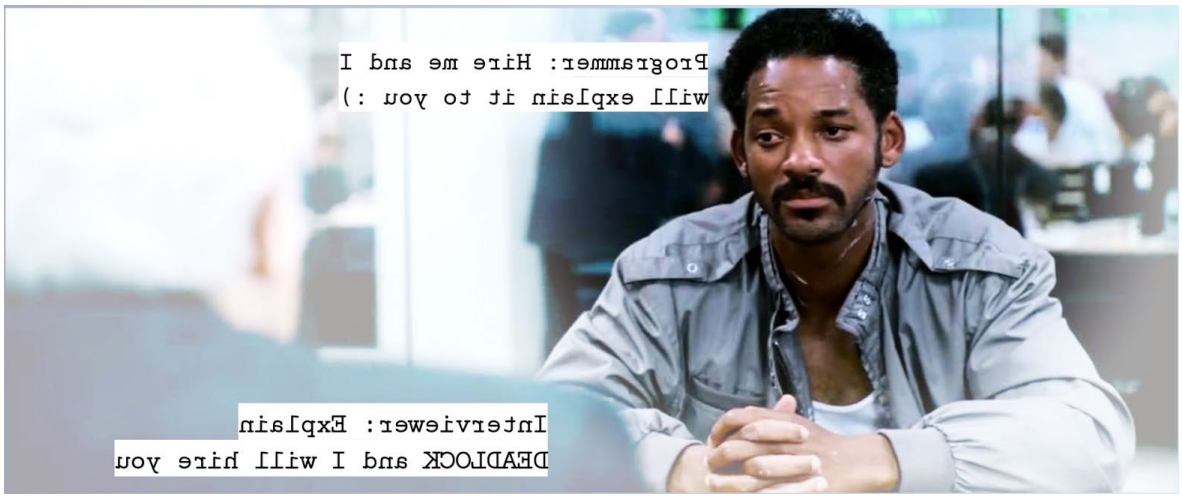
Create your own copy of this template to edit.

In the menu, click **File > Make a copy...**



مقدمه

در این تمرین به تحلیل داده هایی که از مشخصات و قیمت فروش گوشی‌های موبایل جمع‌آوری شده‌است پرداخته شده است. در ابتدا برنامه اقدام به خواندن و تجزیه مجموعه داده[1]ی ارائه شده می‌کند و آنها را در حافظه خود ذخیره می‌کند. پس از استخراج داده‌ها و ویژگی‌های آنها، برنامه اقدام به نرمال‌سازی[2] 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 اوّل

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

جواب: هر پیاده‌سازی سری را می‌توان به عنوان یک عملیات چند مرحله‌ای نگاه کرد که در هر مرحله آن فقط یک عمل می‌تواند انجام پذیرد. این مرحله به مرحله بودن، باعث راحتی در خرد کردن برنامه و تفکیک بخش‌های مختلف با وظایف متفاوت می‌شود. به عنوان مثال، در همین پروژه لایه اولی که می‌توان برنامه را با آن شروع کرد، خواندن ورودی و ذخیره‌سازی آن در حافظه برنامه است. لایه دومی که می‌توان آن را تقسیم کرد، پیدا کردن مینیمم و ماکزیمم است، لایه سوم نرمال‌سازی و دسته چهارم دسته‌بندی داده‌ها (ابتدا ضرب داخلی سپس مشخص کردن دسته آن داده) است. در نهایت وقتی قالب کلی برنامه (در پیاده‌سازی سری) مشخص شود، می‌توان با توجه به معیارهای سنجش موازی‌سازی، بخش‌هایی که امتیاز لازم را با توجه به این معیارها کسب می‌کنند، با استفاده از متد Multithreading پیاده‌سازی کنیم.

سوال دوم

با بررسی زمان اجرای بخش‌های مختلف برنامه، **Hotspot**[۳] های برنامه را مشخص کنید.

جواب:با توجه به اینکه برنامه در پنج بخش اصلی اجرا می‌شود، زمان اجرای میانگین هر پنج قسمت را به دست آوردیم که به شرح زیراند:

- خواندن داده‌ها

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اوّل
٦٤٣٥,٥	٦٣٩٣	٦٤٢٣	٦٥٦٧	٦٧١١	٦٢٤٧	٦٢٧٨

- یافتن اکسترمم‌ها

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اوّل
٦٦٢	٧٧٤	٦٤٤	٧٢١	٥٧٢	٥٨٠	٦٨١

- نرمال‌سازی

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اوّل
٨٠٢	٧٤٣	٨١٥	٧٨٠	٨٤٤	٨١٥	٨١٦

[۳] توابعی که در برنامه‌تان بیشترین زمان اجراها را به خود اختصاص می‌دهند.

- دسته‌بندی

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اوّل
۳۲٤٥,٥	۳٤۲۱	۳۲۹۸	۳۳۷٥	۳۲۲۹	۳۱۷۲	۳۰۸۰

- محاسبه دقت

میانگین	اجرای ششم	اجرای پنجم	اجرای چهارم	اجرای سوم	اجرای دوم	اجرای اوّل
٤۷	٤۷	٤٦	٤۷	٤۷	٤۹	٤٦

با توجه به مقادیر به دست آمده، مورد اول و چهارم، Hotspotهای برنامه هستند که آنها را به صورت Multithread پیاده‌سازی خواهیم کرد.

جدول اوّل

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 اوّل
۱۱۲۷٤,٥	۱۰۹۹٥	۱۱٤۳۳	۱۱٤۷۰	۱۱۷۲۳	۱۱۱۲٦	۱۰۹۰۰

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؟

جواب: در این حالت چند راهکار هست که قابل اجرا می‌باشند:

- کاهش موارد استفاده از mutex_block.
- تغییر تعداد Threadهای استفاده شده که با توجه به ویژگی‌های سخت‌افزاری سیستم می‌توان از این راهکار استفاده کرد؛ لزوما تعداد Thread زیاد موثر نخواهد بود.
- استفاده از ساختارهای داده بهینه‌تر؛ برای مثال با توجه به مشاهدات بنده، آرایه، لیست و وکتور ساختمان داده‌های بهتری نسبت به map، dictionary و .. هستند و سرعت search در آنها بالاتر است.

سوال چهارم

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

جواب: در هنگام پیاده‌سازی بخش خواندن از فایل‌ها، مشاهده کردم که خواندن از فایل نه تنها سریع‌تر انجام نمی‌شود بلکه کندتر هم هست؛ با کمی بررسی متوجه شدم که علت آن استفاده از mutex_lock هنگام نوشتن در حافظه برنامه (vector) بود؛ برای رفع آن برای هر Thread، یک vector محلی گرفته و در انتها آن‌ها را در یک vector بزرگتر global اضافه می‌کنیم. قابل مشاهده است که تا حد خوبی مشکل حل شد.

سوال پنجم

با توجه به تجربه‌ای که در پیاده‌سازی این تمرین بدست آوردید، به نظر شما در چه مواقعی از قفل[4] در یک طراحی چندریسه‌ای ضروری است؟ تاثیر استفاده از قفل‌ها را بر روی کارآیی[5] سامانه بیان کنید.

جواب: با توجه به اینکه استفاده از قفل‌ها تاثیر بسیار زیادی بر کارایی پیاده‌سازی multithread دارد و عملا هنگام lock شدن بخش زیادی از کاربرد موازی‌سازی بی‌اثر می‌شود، بهتر است که موارد استفاده از آن را در برنامه به حداقل برسانیم. اما عملا گاهی استفاده از آن‌ها اجتناب‌ناپذیر است و باید مورد استفاده قرار بگیرند؛ از جمله این موارد، می‌توان به زمان‌هایی که تمامی ریسه‌ها می‌خواهند در یک خانه حافظه بنویسند و یا آن را تغییر دهند اشاره کرد.

---

[4] Lock

[5] Performance

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 اوّل
۷۶۱۳,۸۳	۷۹۷۵	۷۸۹۰	۷۰۶۳	۷۰۳٤	۸۸۳۱	۶۸۹۰

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

میزان تسریع	میانگین زمان اجرای موازی	میانگین زمان اجرای سری
۰,٦۷٥۳۱	۸۸۳۹,۱٦	۱۱۲۷٤,٥