به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی کامپیوتر

Multithreaded Index Mapper with Collision Handling

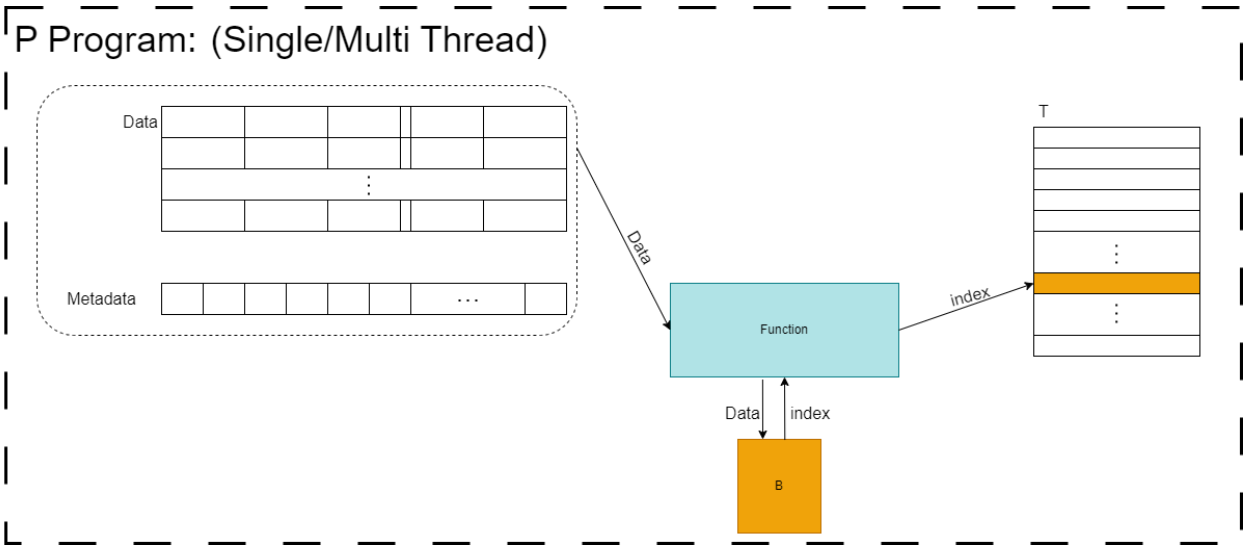
تمرین شماره ۲ درس «پردازش چندهسته‌ای»

استاد: سید مهدی ابراهیمی

نیمسال دوم ۱۴۰۴-۱۴۰۳

قسمت ۱: مقدمه

در ادامه‌ی سؤال «Multithreaded Index Mapping with Collision Handling» که در آزمون میان‌ترم کلاس مطرح شده بود، در این تمرین قصد داریم عملکرد برنامه‌ی شما را توسعه دهیم تا از عملیات حذف (Delete) نیز پشتیبانی کند. با توجه به شکل ۱ هدف در سوال «Multithreaded Index Mapping with Collision Handling» طراحی برنامه‌ای بود که با استفاده از فراداده‌ی داده‌ها، برای هر داده یک index یکتا در جدول T تولید کند؛ به‌گونه‌ای که تکراری‌ها به یک index مشترک نگاشت شوند و collision ها به‌درستی مدیریت شوند.



شکل ۱: یک شماتیک کلی از مسئله

در این تمرین، می‌خواهیم قابلیت حذف را به این سیستم اضافه کنیم.

قسمت ۲: پشتیبانی از عملیات حذف داده‌ها

- در این قسمت باید عملیات حذف را به اپلیکیشن خود اضافه کنید.
- شما باید برنامه‌ای با بالاترین کارایی (Performance) طراحی و پیاده‌سازی کنید که ضمن تولید index برای تمامی داده‌ها و مدیریت کامل برخوردها (Collision Handling)، قابلیت حذف ایمن و هم‌زمان داده‌های مشخص‌شده را نیز پشتیبانی کند.
- همچنین عملیات درج باید وجود یا عدم وجود یک داده را به همراه index به خروجی برگرداند. به صورت زیر:

431,F

- در اینجا عدد ۴۳۱ ایندکس محتوای درج شده و F نشان دهنده عدم وجود این داده در این درج است.
- برای حذف یک Data از جدول باید ابتدا مقدار Data را جستجو کرده و درصورتی که این مقدار موجود بود، آن را حذف کنید. اگر عملیات حذف با موفقیت انجام شد باید یک مقدار True به عنوان درست برگرداند در غیر اینصورت باید مقدار False برگردانده شود.

- حذف باید به‌صورت thread-safe پیاده‌سازی شود و نباید موجب race condition یا رفتار نامشخص شود.
- الزامات اصلی برنامه‌ی P عبارت‌اند از:
  - کارایی بسیار بالا (High Performance)
  - مدیریت برخوردهای احتمالی (Collision Handling)

قسمت ۳:

- پس از تکمیل برنامه، آن را تحت پیکربندی‌های مختلف (Configurations) اجرا کنید و نتایج حاصل را در قالب نمودار ارائه دهید.

جدول۱: مشخصات اجرا

کانفیگ	توضیحات
اجرا به ازای تعداد داده‌های کل: 150K و 300K, 600K	این داده‌ها در فایل‌هایی است که در اختیار شما داده شده است. از هر تعداد 2-set در اختیار شما قرار می‌گیرد.
اجرا به ازای تعدادی نخ‌های 2^0 الی 2^{10}	این پارامترها را می‌بایست در آرگومان برنامه تعریف و به ازای مقادیر مختلف، اجرا کنید.
در نظر گرفتن سایز جدول T به ابعاد ۳ و ۴ و ۵ برابر تعداد کل داده‌های متمایز	این پارامترها را می‌بایست در آرگومان برنامه تعریف و به ازای مقادیر مختلف، اجرا کنید.

- در هر فایل داده، تعداد داده‌های متمایز (Unique) دقیقاً یک‌پنجم از تعداد کل داده‌ها است. برای مثال، اگر تعداد کل داده‌ها ۶۰۰ هزار (۶۰۰K) باشد، تعداد داده‌های متمایز برابر با ۱۲۰ هزار (۱۲۰K) خواهد بود.
- برنامه باید به‌گونه‌ای طراحی شود که پیکربندی‌ها از طریق آرگومان‌های خط فرمان (Command Line Arguments) دریافت شوند.
- **نمونه‌ای** از نحوه اجرای برنامه با آرگومان‌های مختلف برای تعداد نخ (Threads):

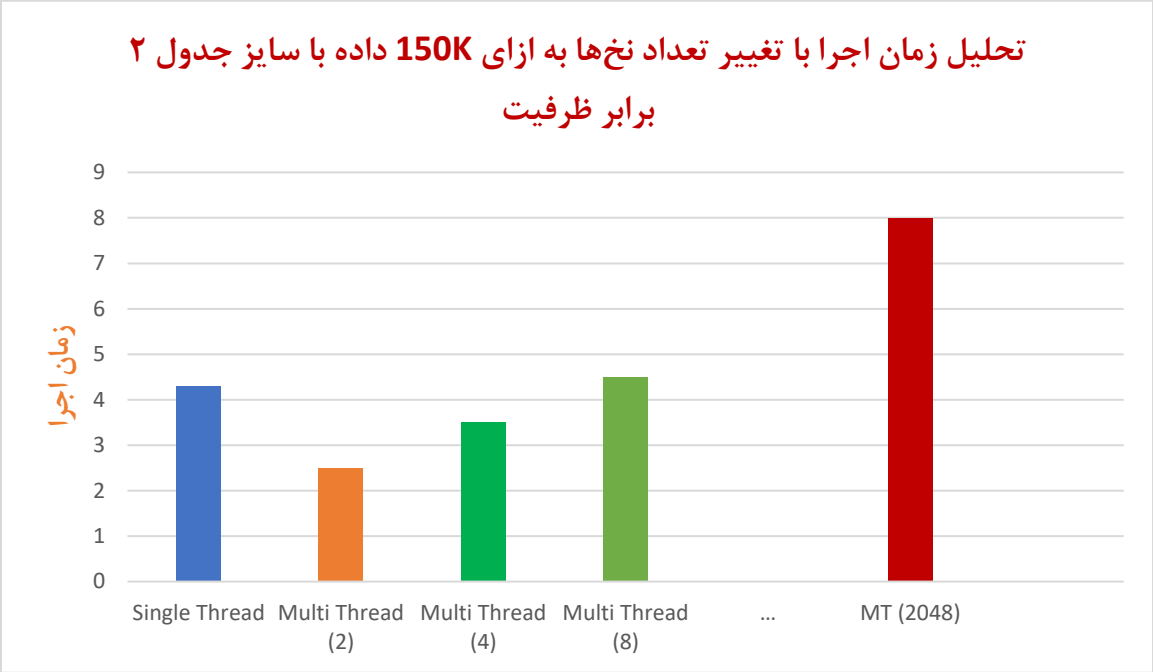
```
My_app --data_size 150K --threads 1 --tsize 150K --flow insert insert delete insert --input 150K_set1.txt 150K_set2.txt 150K_set1.txt 150K_set2.txt
```

```
My_app --data_size 150K --threads 2 --tsize 150K --flow insert insert delete insert --input 150K_set1.txt 150K_set2.txt 150K_set1.txt 150K_set2.txt
```

```
My_app --data_size 150K --threads 8192 --tsize 150K --flow insert insert delete insert --input 150K_set1.txt 150K_set2.txt 150K_set1.txt 150K_set2.txt
```

- توجه داشته باشید که تعداد ورودی‌ها و تعداد عملیات‌ها باید برابر باشند.

نحوه ورودی‌ها می‌بایست دقیقا مطابق با الگوی ذکر شده باشد.



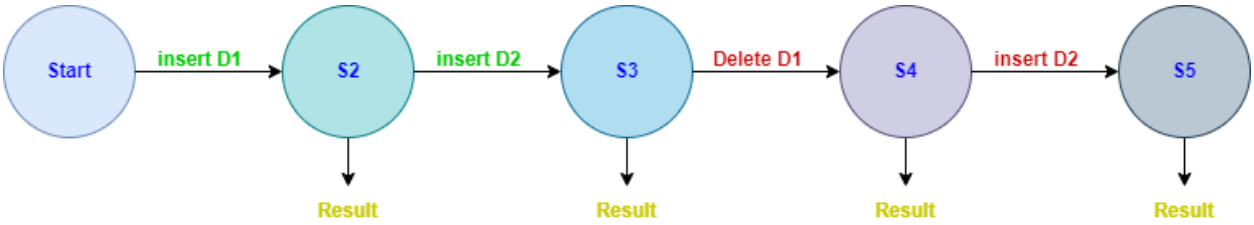
نمودار ۱: نشان دهنده زمان اجرای یک عملیات برای تعداد نخ‌های مختلف

انتظار می‌رود برای اجراهایی متناسب با جدول۱ ، ۹ مجموعه نمودار که هرکدام مشابه نمودار بالا هستند،تولید شود.

- پس از پیاده‌سازی برنامه، آن را مطابق روند توضیح داده شده در زیر و نشان داده‌شده در شکل۲ تست کنید (به عنوان مثال برای مجموعه داده ۱۵۰K این روند را می‌خواهیم انجام دهیم):
  ۱- مجموعه داده‌ی اول با نام 150K_set1.txt را در جدول درج و index ها را در فایل Result ذخیره کنید.
  ۲- مجموعه داده دوم با نام 150K_set2.txt را در جدول درج و index ها را در فایل Result ذخیره کنید.
  ۳- در مرحله‌ی بعد، مجموعه‌ی اول (150K_set1.txt) را حذف کرده و index های حذف‌شـــده را در فایل Result ذخیره کنید.
  ۴- در نهایت، مجدداً مجموعه‌ی دوم (150K_set2.txt) را در جدول درج کرده و indexهای آن را در فایل Result ثبت نمایید.
- الزامات مربوط به ذخیره‌سازی خروجی:
  - تمامی فایل‌های خروجی مربوط به مراحل بالا باید در پوشه‌ی result/ ذخیره شوند.
- نکات مربوط به صحت عملکرد:

۱- تطابق indexها: پس از حذف مجموعه‌ی اول، indexهایی که در مرحله‌ی ۳ (حذف دیتاست 150K_set1 از جدول) در در فایل Result درج می‌شوند، باید دقیقاً با indexهایی که در مرحله‌ی ۱ (درج دیتاست 150K_set1 در جدول) تولید شده بودند، یکسان باشند.

۲- جلوگیری از درج مجدد: هنگام درج مجدد مجموعه‌ی دوم (مرحله ۴)، برنامه نباید داده‌ی تکراری را در جدول درج کند؛ بلکه باید با شناسایی وجود داده، عملیات درج را نادیده گرفته و مقدار index و True را بازگرداند.



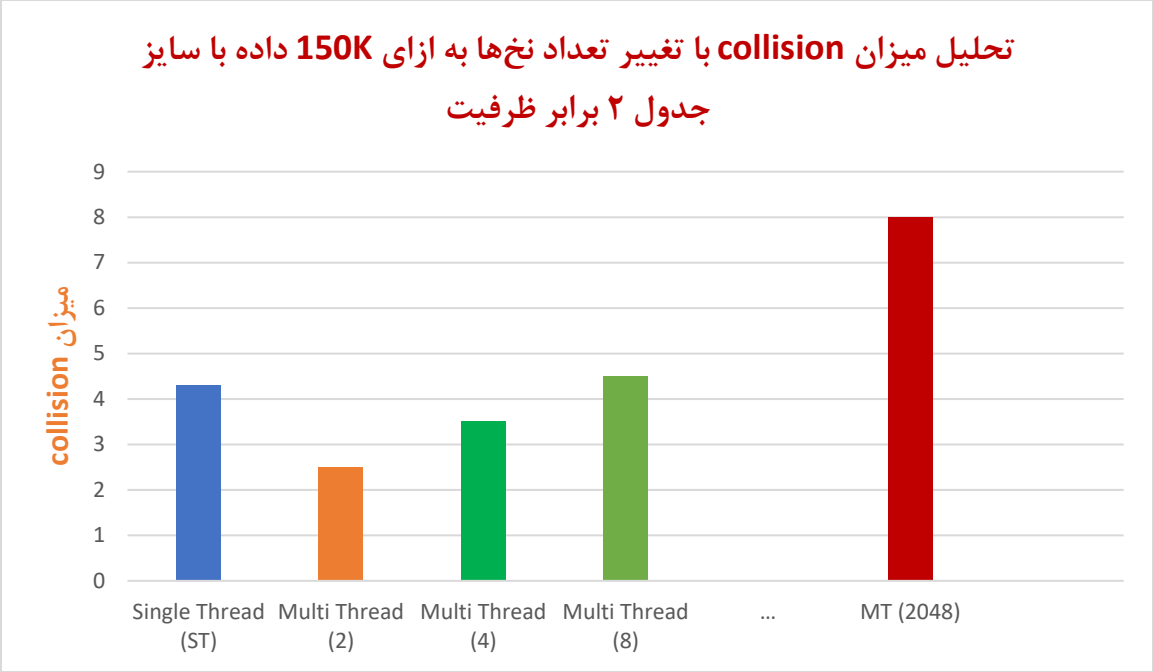
شکل۲: روند تست عملیات حذف از جدول

- توجه داشته باشید که ساختار نمودارهای خروجی باید مطابق قالب مشخص‌شده در فایل نمونه باشد. هرگونه نمودار با قالب متفاوت بررسی نخواهد شد.
- شما موظف هستید تمامی Dataها و indexهایی که توسط برنامه تولید می‌شوند را در یک فایل Result ذخیره کنید. فرمت محتوایی این فایل باید به‌صورت زیر باشد (T: یعنی این داده از قبل در جدول وجود داشته است. F: یعنی این داده از قبل در جدول وجود نداشته است و در عملیات حذف در صورتی که Data در جدول وجود نداشت، index را قرار ندهید):
  - Data:Index1:T, Data:Index2:F, Data:Index3:T,…
  - index ها باید به همان ترتیبی که در آرایه‌ی فراداده (از لندیس ۰) آمده‌اند ثبت شوند. خروجی باید متن ساده (plain-text) و قابل خواندن برای انسان باشد، و مقادیر با کاراکتر کاما «,» از یکدیگر جدا شده باشند.
- الزامات کلیدی مربوط به indexهای تولیدشده:
  - حفظ الگوی تکرار: از آنجا که مجموعه داده‌ها از پیش مشخص شده‌اند، محل و الگوی داده‌های تکراری نیز قابل پیش‌بینی است.در نتیجه، انتظار داریم indexهایی که برای داده‌های تکراری تولید می‌شوند نیز دقیقاً همان الگو و ترتیب تکرار داده‌ها را منعکس کنند.
  - درستی تعداد indexهای متمایز: تعداد indexهای یکتا (متمایز) باید دقیقاً برابر با تعداد داده‌های یکتا باشد.با توجه به اینکه می‌دانید چه داده‌هایی تکراری هستند، می‌توانید با استفاده از روش‌هایی مانند assert در برنامه، این انطباق را بررسی کرده و از صحت عملکرد خود اطمینان حاصل کنید.
- توصیف سیستم اجرای آزمایش‌ها نیز الزامی است و باید شامل مشخصات جدول۲ باشد:

جدول۲: توصیف سیستم اجرای آزمایش‌ها (**برای مثال**)

پردازنده (CPU)	Intel Core i7 – 8 Cores @ 2.0 GHz
حافظه (RAM)	16 GB DDR4 @ 4800 MT/s
سیستم‌عامل (OS)	Ubuntu Server 24.04 (64-bit)

- تمامی اجرای آزمایش‌ها باید بر روی یک سیستم ثابت انجام شوند و همچنین یک جدول شامل مشخصات کامل این سیستم نیز ارائه گردد.
- **پرسش۱**: بهترین زمان اجرا به ازای چند thread بدست می‌آید؟
- **پرسش۲**: افزایش تعداد threadها از چه تعدادی به بعد، زمان اجرا را از ST بدتر می‌کند؟ چرا؟ (ارائه تحلیل الزامی است)



انتظار می‌رود که با تغییر تعداد نخ‌ها، میزان برخورد (collision) ثابت باقی بماند؛ با این حال، ممکن است برنامه‌ی شما رفتار متفاوتی نشان دهد. از این رو، لازم است میزان collision را برای مقادیر مختلف نخ‌ها محاسبه کرده و در قالب نمودار گزارش دهید. در مجموع، باید ۹ نمودار مشابه نمونه‌های ارائه‌شده تولید شود (برای ترکیب ۳ مقدار متفاوت تعداد کل داده‌ها و ۳ اندازه‌ی مختلف جدول T).

قوانین:

- کد توسعه داده‌شده توسط شما می‌بایست توسط makefile قابل build باشد.
- کد شما صرفا توسط ماشین build می‌شود و می‌بایست توسط ماشین قابل build باشد.
- برنامه‌ی شما صرفا توسط ماشین اجرا می‌شود.
- نتایج برنامه‌ی شما می‌بایست الزاما در ساختار مشخص شده خروجی تولید کند. نتایج صرفا با ماشین بررسی می‌شوند.

هشـــدار: تولید indexها به صـــورت اســـتاتیک، یعنی تولید indexها بدون هیچگونه پردازش و صـــرفا بر اســـاس داده‌های آپلود شـده، فاقد هیچ‌گونه ارزش و منجر به نمره 0 برای تمرین خواهد بود. در نظر داشـته باشـید که داده‌های آپلود شده توسط ما می‌توانند در زمان تصحیح، تغییر کنند و این مسئله نباید هیچ آسیبی به تولید نتایج جدید شما وارد کند. انتظار داریم تحلیل‌ها و حدود نتایج زمان‌اجرا، مطابق با گزارش‌های شما باشد.

ساختار فایل تحویلی

- برنامه‌ی شما توسط سامانه به‌صورت خودکار از حالت فشرده خارج خواهد شد. لطفاً فایل نهایی را با فرمت .tar.gz و با استفاده از دستور زیر ایجاد کنید:

```
tar -czf  HW2_MCC_030402_StudentID.tar.gz HW2_MCC_030402_StudentID
```

- ساختار پوشه‌ی داخلی باید دقیقاً به صورت زیر باشد:
  - پوشه‌ی HW2_MCC_030402_StudentID
    - پوشه‌ی bin شامل باینری شما
    - پوشه‌ی src شامل کدهای شما
    - پوشه‌ی results، شامل نتایج شما
  - برنامه‌ی باینری، می‌بایست با فرمت زیر تولید شود. صرفا توسط ماشین اجرا می‌شود.
    - HW2_MCC_030402_StudentID

ساختار تولید نتایج و فایل‌های خروجی

- نام فایل‌های نتایج باید مطابق الگوی زیر باشد:

```
Results_ HW2_MCC_030402_StudentID_{DataSize}_{NumOfThreads}_{TSize}_{flow}.txt
```

  - به‌عنوان مثال:

```
Results_ MCC_030402_StudentID_150K_512_300K_insert_insert_delete_insert.txt
```

- تولید سایر فایل‌های لاگ (به جز فایل‌های نتایج) آزاد است و به دلخواه شما انجام می‌شود؛ این فایل‌ها در فرآیند تصحیح ملاک ارزیابی قرار نخواهند گرفت.

محتوای مورد انتظار برای هر فایل نتایج

- هر فایل نتیجه باید دقیقاً با فرمت زیر باشد.

Actions: insert

ExecutionTime: 100 ms

NumberOfHandledCollision: 1122

Data:index1:T, Data:index2:F, Data:index3:F…

Actions: insert

ExecutionTime: 160 ms

NumberOfHandledCollision: 1642

Data:index1:T, Data:index2:F, Data:index3:F…

Actions: delete

ExecutionTime: 100 ms

NumberOfHandledCollision: 1341

Data:index1:T,Data:F,Data:F…

…

- مقادیر بالا به‌صورت نمونه هستند؛ برنامه باید مقدار واقعی زمان اجرا و تعداد برخوردهای مدیریت‌شده را تولید کند.
- لطفاً از هرگونه محتوای اضافی یا تغییر در ساختار خطوط فوق خودداری فرمایید.

۱. مراحل انجام تمرین باید به‌صورت گزارش ارائه شود. گزارش باید شامل نتایج به‌دست آمده و سایر موارد خواسته شده به‌صورت ذکر شده درصورت پروژه باشد.

۲. الزامات فنی: حتما از زبان برنامه‌نویسی C استفاده کنید. همچنین کد شما باید روی سیستم عامل Ubuntu Linux (نسخه 22.04) کامپایل و اجرا شود و برای مدیریت ریسمان‌ها حتما از کتابخانه pthread استفاده شود.

۳. فایل‌ها، خروجی‌های به دست آمده (کد برنامه، library، makefile و نسخه باینری اپلیکیشن تست و ...) و فایل گزارش را به‌صورت فشرده با فرمت زیر در سامانه درس‌افزار (CW) بارگزاری نمایید.

HW2_MCC_030402_StudentID.tar.gz

۴. تاریخ تحویل تمرین ۲۰ خرداد است و این تاریخ به هیچ‌وجه تغییر نمی‌کند و به ازای هر روز تاخیر ۱۵٪ نمره را از دست خواهید داد و بعد از ۳ روز نمره این تمرین ۰ خواهد شد.

۵. می‌توانید سوالات یا ابهامات خود را به ایمیل s.yazdan566@gmail.com ارسال نمایید.

۶. رعایت آداب آموزشی در انجام پروژه و تمرین‌های درسی الزامی است. لطفا آیین‌نامه مصوب دانشکده (آداب‌نامه‌ی انجام تمرین‌های درسی) را دقیقا مطالعه فرمایید. در صورت مشاهده هرگونه تقلب علمی، نمره تمرین برای هر دو طرف ۱۰۰- منظور خواهد شد.

با آرزوی موفقیت