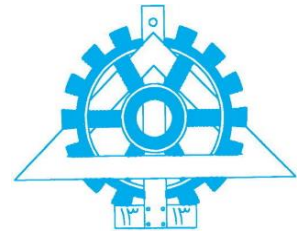




باسمه تعالی سیستم‌های عامل پروژه‌ی دوم درس تاریخ تحویل: ۲۵ آبان



هدف از این تمرین، آشنایی با نحوه مدیریت پروژه‌ها و راه‌های ارتباطی میان آن‌ها می‌باشد. در این تمرین با بهره‌گیری از عملیات‌هایی در سطح پروژه‌ها محاسباتی را روی داده‌هایی از اجناس فروشگاه انجام خواهید داد.

شرح پروژه

در این پروژه قرار است دو عمل یافتن کمینه^۲ و یافتن بیشینه^۳ را روی داده‌هایی به صورت موازی پیاده‌سازی کنیم. مجموعه داده‌ای از قیمت اجناس مختلف در روزهای مختلف که توسط فروشگاه‌های زنجیره‌ای در شهرهای برخی استان‌ها قرار دارند را در اختیار داریم. می‌خواهیم دو عمل بیان شده را برای قیمت روی اجناسی که توسط این فروشگاه‌های زنجیره‌ای در شهرهای استان‌های مختلف کشور در تاریخ‌های خاصی عرضه شده‌اند انجام دهیم.

نحوه پیاده‌سازی به این شکل است که در ابتدا یک پروژه اولیه وجود دارد که دستورات را از یک پرونده^۴ که آدرس آن هنگام اجرای برنامه داده می‌شود، خوانده و سپس به ازای هر یک از دستورات یک پروژه جدید ایجاد می‌کند که وظیفه به‌دست‌آوردن خروجی آن دستور را دارد. هر پروژه سطح دوم (یعنی پروژه‌هایی که توسط پروژه اولیه ساخته می‌شوند) به ازای هر یک از استان‌های موجود در پوشه^۵ مربوطه یک پروژه ساخته و هر کدام از آن پروژه‌های جدید نیز به ازای هر پوشه موجود در آن که مشخص‌کننده شهر می‌باشد یک پروژه جدید می‌سازد. در انتها در پوشه شهر ممکن است تعدادی پرونده موجود باشد که هر کدام شامل اطلاعات یک فروشگاه خاص است. به ازای هر کدام از آن‌ها نیز یک پروژه می‌سازیم که پردازش اطلاعات روی آن پرونده را انجام دهد. همچنین ممکن است در شهر مذکور هیچ فروشگاه‌ای نباشد، در این صورت پروژه آن شهر باید خاتمه یابد. این پروژه‌های جدید روی پرونده‌های داده‌های مربوط به آن فروشگاه‌ها، عمل مذکور را اجرا کرده و خروجی خود را به پروژه پدر خود می‌دهند و خاتمه می‌یابند. خاتمه‌یافتن به موقع این پروژه‌ها مهم می‌باشد. به این شکل یک درخت از پروژه‌ها داریم.

هر پردازنده پدر برای انتقال اطلاعات لازم از خود به پردازنده فرزند باید از Unnamed Pipe استفاده کند. برای انتقال خروجی از پردازنده فرزند به پردازنده پدر باید از Named Pipe استفاده شود تا اطلاعات از این طریق از طرف پروژه فرزند در اختیار پروژه

¹ Process

² Minimum

³ Maximum

⁴ File

⁵ Directory

پدر قرار گیرد. به عبارت دیگر در درخت پردازشهایی که به وجود می‌آید انتقال اطلاعات از بالا به پایین (پدر به فرزند) توسط Unnamed Pipe و انتقال اطلاعات از پایین به بالا (فرزند به پدر) توسط Named Pipe صورت خواهد گرفت.

در نهایت پردازش اولیه باید نتیجه هر دستور را در خروجی استاندارد چاپ کند. توجه شود که خروجی‌ها باید به ترتیب دستورها باشند. یعنی برای مثال خروجی دستور اول در خط اول خروجی و خروجی دستور دوم در خط دوم خروجی نمایش داده خواهد شد.

فرمت داده‌های فروشگاه‌ها

برای هر فروشگاه در شهر یک پرونده در قالب CSV موجود است که داده‌های مربوط به کالاهای عرضه‌شده توسط آن فروشگاه در آن ذخیره شده است. فرمت این داده‌ها به شکل زیر است که در آن `<date>` تاریخ عرضه کالا، `<product_id>` شناسه کالا و `<price>` قیمت کالا را مشخص می‌سازد که عددی صحیح و مثبت است.

`<date>`, `<product_id>`, `<price>`

فرمت تاریخ به شکل YYYY/MM/DD است. به طور مثال تاریخ برای روز اول ماه دوم سال ۲۰۲۰ به شکل 2020/02/01 نمایش داده می‌شود. همچنین توجه داشته باشید در یک فروشگاه می‌تواند بیش از یک کالا با شناسه‌ای مشخص وجود داشته باشد که به طور مثال می‌تواند نشان‌دهنده قیمت آن کالا با تاریخ عرضه دیگری باشد.

قالب دستورها

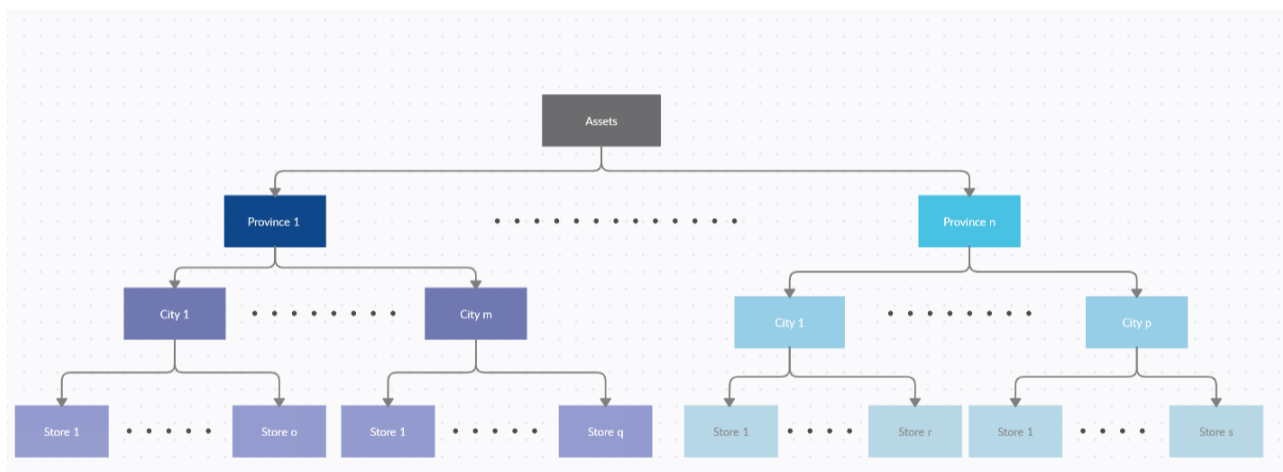
قالب دستورها به شکل زیر می‌باشند. دستور اول برای یافتن بیشینه قیمت و دستور دوم برای یافتن کمینه قیمت محصول با شناسه محصول `<product_id>` در بین داده‌هایی با این شناسه است که تاریخ عرضه بین دو تاریخ `<starting_date>` و `<ending_date>` را داشته باشند. برای بررسی بین دو تاریخ، خود تاریخ‌های شروع و پایان را نیز در نظر بگیرید.

MAX `<product_id>` `<starting_date>` `<ending_date>`

MIN `<product_id>` `<starting_date>` `<ending_date>`

نحوه پوشه‌بندی داده‌ها

همان طور که ذکر شد یک پوشه اصلی برای داده‌ها وجود دارد که ساختار پوشه‌های موجود در آن به شکل زیر است:



همان‌طور که دیده می‌شود در سطح اول استان‌ها را داریم. در هر استان چند شهر داریم و در هر شهر چند فروشگاه داریم. توجه شود که عمق تمام شاخه‌های یک درخت لروما با یکدیگر برابر نیست یعنی با توجه به ساختار بالا یک شهر ممکن است فروشگاه‌های نداشته باشد و یا یک استان شهری نداشته باشد. همچنین توجه داشته باشید لزومی وجود ندارد که تعداد فروشگاه‌ها در شهرها برابر باشد. ممکن است در شهر n به تعداد k فروشگاه و در شهر m به تعداد l فروشگاه داشته وجود داشته باشد.

ورودی و خروجی برنامه

برنامه شما باید به ترتیب آدرس مربوط به پرونده دستورات و پوشه دربرگیرنده داده‌های فروشگاه را از طریق آرگومان‌هایی در رابط خط فرمان^۶ دریافت کند. همچنین نیاز است نام نهایی برنامه قابل اجرا `StoreCalculator.out` باشد. به طور مثال نمونه ورودی برای برنامه با فرض این که پرونده دستورات با نام `commands.txt` و پوشه داده‌ها با نام `assets` در کنار پرونده اجرایی شما قرار گرفته باشد به شکل زیر است:

```
./StoreCalculator.out commands.txt assets
```

خروجی برنامه شما باید در هر خط به ترتیب پاسخ دستور داده‌شده در پرونده دستورات باشد. در صورتی که برای دستوری پاسخ وجود نداشت مقدار 1- چاپ شود.

نکات پایانی

- برنامه شما باید با استفاده از `Makefile` و با مترجم `g++`^۷ و با استاندارد `C++11` ترجمه شده و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- دقت کنید هدف پروژه طراحی و استفاده درست از مفاهیم موازی‌سازی پردازش‌هاست و سایر پیاده‌سازی‌ها قابل قبول نیستند.
- برای تحویل پروژه می‌توانید از یکی از دو روش زیر استفاده کنید:
 - تمامی نتایج را در یک پرونده فشرده شده با عنوان `OS-CA2-<#SID>.zip` در محل بارگذاری در سایت درس آپلود کنید.
 - با توجه به مشکلات پیش آمده برای استفاده از `GitLab` توسط `ip` های ایران، امکان استفاده از این سایت برای آپلود پروژه وجود ندارد. اما می‌توانید از سایت `bitbucket` به جای آن استفاده کنید و مانند `GitLab` پروژه خود را پیش ببرید. همچنین باید اکانت `os_ta_fall2020@yahoo.com` را به عنوان یکی از اعضا به مخزن خود اضافه کنید. در انتها کافی است در محل بارگذاری در سایت درس، آدرس مخزن و شناسه آخرین `commit` خود را آپلود کنید.
- توجه شود که پروژه‌های درس تک‌نفره‌اند و پروژه باید برای دانشجویان مهندسی کامپیوتر به صورت انفرادی و برای دانشجویان غیر مهندسی کامپیوتر در قالب گروه سه نفره انجام شود.

^۶ Command Line Interface

^۷ Compiler

- نکاتی که در جلسه توجیهی و یا فروم درس مطرح می‌شود بخشی از صورت پروژه هستند لذا به شما توصیه می‌شود که حتما در جلسه توجیهی شرکت کنید.
- در صورت وجود ابهام یا پرسش از طریق ایمیل های زیر با تی ای های مربوطه در ارتباط باشید.

- 1- a94h.73@gmail.com
- 2- arman.rostami.999@gmail.com
- 3- azarenejad99@gmail.com