



پروژه دوم درس



سیستم های عامل - بهار ۱۴۰۲

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراحان تمرین:

۱۴۰۲/۰۱/۲۷ ساعت ۲۳:۵۹

استاد: دکتر مهدی کارگهی

محمد فرهی، سروش صادقیان

مقدمه

در این پروژه می‌خواهیم با نحوه مدیریت کردن پردازش‌ها و روش‌های ارتباطی آن‌ها آشنا شویم. در این تمرین با بهره‌گیری از عملیات‌هایی در سطح پردازش‌ها محاسباتی را روی داده‌هایی از سن ورزشکاران فوتبال انجام خواهید داد.

مدل نگاشت کاهش^۲

در دنیای امروز، به دلیل گسترش اینترنت و دستگاه‌های هوشمند، روزانه حجم زیادی از داده تولید می‌شود. در گذشته، داده‌های تولیدی قابلیت ذخیره و اجرا بر روی یک دستگاه سخت‌افزاری را داشتند اما امروزه برای بسیاری از موارد این امر غیر ممکن است. نگاشت کاهش یک چارچوب و مدل برنامه‌نویسی است که اجازه اجرای

^۱ process

^۲ Map-Reduce

پردازش موازی و توزیع شده بر روی مجموعه بزرگی از داده‌ها در یک محیط توزیع یافته را می‌دهد و بسیاری از

مفاهیم آن از زبان‌های تابع‌گرا³ مانند Lisp گرفته شده‌است. در ادامه به توضیح این مدل خواهیم پرداخت:

Map-Reduce از دو بخش نگاشت و کاهش تشکیل می‌شود.

● در قسمت نگاشت، تعدادی پردازش برای عمل نگاشت وجود دارد که کاملاً مستقل از یکدیگر عمل

می‌کنند و هیچ ارتباطی با یکدیگر ندارند. خروجی این مرحله تعدادی key-value خواهد بود که برای

استفاده به قسمت کاهش ارسال می‌شود. تعداد پردازش‌های قسمت نگاشت محدودیت خاصی ندارد و

می‌تواند بر اساس منابع در دسترس و نوع داده‌ها انتخاب شود. هر کدام از پردازش‌های قسمت نگاشت

می‌توانند به صورت موازی اجرا شوند.

● در قسمت کاهش، خروجی‌های مرحله قبل به عنوان ورودی دریافت می‌شود و سپس بر اساس کلید،

داده‌ها تقسیم می‌شوند. داده‌هایی که کلید یکسان دارند، حتماً باید به یک پردازش داده شوند. هر پردازش

کاهش، بر روی مجموعه داده‌های با کلید یکسان، عملیات موردنظر را انجام می‌دهد و خروجی را

ایجاد می‌کند.

شرح پروژه

در این پروژه شما قرار است به روش نگاشت-کاهش، یک سری شاخص‌های آماری از داده‌های سن و پست⁴

بازیکنان فوتبال به دست آورید. مجموعه داده‌ای از بازیکنان فوتبال باشگاه‌های کشورهای مختلف به همراه سن و

³ Functional

⁴ position

پست آن‌ها در زمین داریم؛ می‌خواهیم کمترین⁵، بیشترین⁶، میانگین⁷ سنی و همچنین تعداد کل بازیکنان برای یک سری پست مشخص را بیابیم.

نحوه پیاده سازی

یک پوشه به نام clubs به شما داده می‌شود که حاوی تعدادی پوشه دیگر با عنوان <country_name> است که هر کدام مربوط به یک کشور است. در هر پوشه مربوط به یک کشور، تعدادی فایل csv مربوط به باشگاه‌های آن کشور قرار دارد. هر فایل csv با نام <club_name.csv> شامل اطلاعات بازیکنان آن باشگاه می‌باشد. به این صورت که در هر خط از فایل، نام یک بازیکن، پست تخصصی آن بازیکن و سن آن بازیکن نوشته شده است. دقت کنید که فرض شده هر بازیکن فقط یک پست تخصصی دارد. برای مثال، PSG.csv می‌تواند به شکل زیر باشد:

Neymar,LW,31

Messi,RW,35

Mbappe,CF,24

...

همچنین در پوشه clubs یک فایل csv دیگر به نام positions.csv وجود دارد که شامل یک خط است و تمام پست‌های موجود در فایل‌های باشگاه‌ها در آن لیست شده است.

⁵ minimum
⁶ maximum
⁷ average

نحوه پیاده‌سازی به این شکل است که در ابتدا یک پردازه اولیه وجود دارد که آدرس پوشه clubs در آرگومان ورودی به آن داده می‌شود. این پردازه به ازای هر پوشه مربوط به کشورها یک پردازه جدید ایجاد می‌کند. سپس در هر پردازه مربوط به یک کشور، به ازای هر باشگاه یک پردازه جدید ایجاد می‌شود.

هر پردازه مربوط به یک باشگاه در حقیقت یک پردازه نگاشت است که با خواندن فایل csv متناظر و دریافت اطلاعات مربوط به سن‌ها و پست‌ها، اقدام به پردازش داده‌ها و فرستان اطلاعات لازم برای محاسبه یک سری شاخص‌ها و اطلاعات آماری به پردازه‌های کاهش می‌کند.

شاخص‌هایی که در این پروژه باید به دست آورده شوند، شامل کمترین سن، بیشترین سن، میانگین سن و تعداد بازیکنان می‌شود. دقت کنید این شاخص‌ها باید برای هر پست و نسبت به کل داده ورودی (پوشه clubs شامل همه باشگاه‌ها) محاسبه شوند. به عنوان مثال به دست آوردن کمترین سن پست دفاع در بین کل بازیکنان باشگاه‌ها مد نظر است نه برای یک باشگاه.

همان‌طور که اشاره شد، پردازه نگاشت مربوط به هر باشگاه، اطلاعات مورد نیاز برای محاسبه شاخص‌های مورد نیاز را برای هر پست در آن باشگاه به دست می‌آورد. سپس این پردازه اطلاعات مربوط به هر پست را به پردازه متناظر آن پست انتقال می‌دهد. به ازای هر پست، یک پردازه جداگانه خواهیم داشت که همان پردازه کاهش است و توسط پردازه اولیه ایجاد می‌شود؛ این پردازه با استفاده از اطلاعاتی که از پردازه هر باشگاه به آن منتقل می‌شود، شاخص‌های مورد نظر را برای آن پست محاسبه کرده و گزارش می‌کند. نام و تعداد کل پست‌ها در ابتدا نامعلوم است و باید با خواندن فایل positions.csv و در پردازه اولیه، این اطلاعات را به دست آورید.

توجه داشته باشید که برنامه شما به صورت تعاملی⁸ است؛ به صورتی که اینکه شاخص‌های مورد نظر برای چه پست‌های گزارش شوند، توسط کاربر مشخص می‌شود. پردازش اولیه شما باید در ابتدای اجرا، تمام پست‌های موجود را از فایل `positions.csv` خوانده و برای کاربر نمایش دهد. کاربر از بین پست‌های موجود، یک یا چند پست را به صورت `space-separated` در ترمینال⁹ برنامه وارد می‌کند. سپس پردازش اولیه بر اساس ورودی کاربر، فرایند ساخت پردازش‌های نگاشت و کاهش را شروع می‌کند و پردازش‌های پست را متناظر با ورودی کاربر می‌سازد.

پس از اتمام پردازش پردازش‌ها و چاپ گزارش، برنامه پایان می‌یابد. دقت کنید که چاپ گزارش و اتمام پردازش‌های فرزند حتما قبل اتمام پردازش اولیه و برنامه اصلی انجام شود.

در بخش ورودی گرفتن از کاربر نیازی به رسیدگی به خطاها یا ورودی‌های نامعتبر نیست و تضمین می‌شود کاربر پست‌های معتبر و با فرمت درست وارد خواهد کرد.

دقت کنید که برای انتقال اطلاعات لازم بین هر دو پردازش پدر و فرزند (مثل انتقال نام پست‌ها به فرزندان) باید از **unnamed pipe** استفاده شود. برای انتقال اطلاعات مورد نیاز برای هر پست از پردازش باشگاه به پردازش متناظر آن پست از **named pipe** استفاده می‌شود (یعنی مثلاً برای انتقال اطلاعاتی مثل تعداد بازیکنان و ... برای پست CF هر باشگاه حتماً باید یک **named pipe** جداگانه بین پردازش مربوط به آن باشگاه و پردازش متناظر با پست CF ساخته شود). دوباره تأکید می‌شود برای انتقال نام پست‌های تحت بررسی به پردازش‌های فرزند(نگاشت یا کاهش) از **unnamed pipe** استفاده می‌شود. در نهایت هر کدام از پردازش‌های متناظر با پست‌ها، باید پس از اتمام پردازش، اطلاعات مورد نظر را چاپ کنند.

⁸ interactive

⁹ terminal

ورودی و خروجی برنامه

برنامه شما باید آدرس مربوط به پوشه clubs را به صورت آرگومان در رابط خط فرمان دریافت کند. همچنین نام

نهایی برنامه قابل اجرا ClubsAgeStats.out باشد. سپس برنامه پس از چاپ پست‌های موجود و گرفتن

پست‌های تحت بررسی از کاربر، پردازش را آغاز می‌کند.

خروجی برنامه شما نیز باید در هر خط نام پست و به ترتیب کمترین سن، بیشترین سن، میانگین سنی و تعداد

بازیکنان مربوط به آن پست را چاپ کند. میانگین سنی را با تقریب یک رقم اعشار چاپ کنید. ترتیب پست‌ها

اهمیت ندارد. نمونه ورودی و خروجی :

```
● $ ./ClubsAgeStats.out clubs
All positions :
CF - LW - RW - AM - CM - DM - CD - RD - LD - GK
Enter positions to get stats :
CF CD
CF min age : 20
CF max age : 35
CF avg age : 26.2
CF count : 30
CD min age : 21
CD max age : 36
CD avg age : 28.1
CD count : 40
○ $ █
```

نکات تکمیلی

- برای ساخت پردازنده‌ها توسط پردازنده اصلی، حتماً از فراخوانی‌های سیستمی `fork` و `exec` برای ساخت و اجرای آنها باید استفاده کنید.
- دقت کنید به ازای هر نوع پردازنده‌ای که در برنامه ایجاد می‌شود، باید حداقل یک فایل `cpp` مربوط به آن پردازنده در فایل‌های پروژه شما وجود داشته باشد منطق آن پردازنده در آن پیاده‌سازی شده است.
- فرمت انتقال داده‌ها میان پردازنده‌ها بر عهده خودتان است.
- بعد از استفاده از پایپ‌ها، آنها را ببندید.¹⁰
- به دلیل استفاده از نوع `pipe` ها در هر مرحله فکر کنید. در زمان تحویل سوالاتی در این باره پرسیده خواهد شد.
- دقت شود تنها راه ارتباطی میان پردازنده‌ها استفاده از `pipe` است و هیچ راه دیگری قابل قبول نیست. اطلاعاتی که از طریق آرگومان به پردازنده‌ها منتقل می‌شود باید بسیار ناچیز باشد.
- هیچ نوع دیگری از پیاده‌سازی بجز مدلی که در بالا توضیح داده شد قابل قبول نیست.

نکات پایانی

- برنامه حتماً باید با استفاده از `makefile` و کامپایلر `g++` اجرا شود.
- برنامه باید در سیستم عامل لینوکس و در زمان معقول اجرا شود.

¹⁰ `close()`

- تمامی نتایج را در یک فایل فشرده با اسم OS-CA2-<#SID>.zip در محل بارگذاری درس آپلود کنید.

- انجام این پروژه به صورت انفرادی است.

- نکاتی که در جلسه توجیهی یا فروم درس مطرح می شوند بخشی از صورت پروژه هستند لذا توصیه می شود که شرکت کنید.

- در صورت داشتن هرگونه سوال با مسئولان این پروژه در ارتباط باشید.

➤ mohammad.farrahi@ut.ac.ir

➤ sadeqiansoroosh@gmail.com