



پروژه دوم درس



سیستم های عامل - پاییز ۱۴۰۲

مهلت تحویل:

دانشکده مهندسی برق و
کامپیوتر

طراحان تمرین:

پویا صادقی، سروش صادقیان

استاد: دکتر مهدی کارگهی

مقدمه

در این پروژه می‌خواهیم با نحوه مدیریت کردن پردازش‌ها^۱ و روش‌های ارتباطی آن‌ها آشنا شویم. در این تمرین با بهره‌گیری از عملیات‌هایی در سطح پردازش‌ها محاسباتی را روی داده‌های بدست آمده از کنتورهای ساختمان‌ها انجام خواهیم 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 غیرممکن است. نگاشت کاهش یک چارچوب و مدل برنامه‌نویسی است که اجازه اجرای پردازش موازی و توزیع‌شده بر روی مجموعه بزرگی از داده‌ها در یک محیط توزیع‌یافته را

^۱ process

^۲ Map-Reduce

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 پردازنده 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 پردازنده‌های قسمت نگاشت محدودیت خاصی ندارد و می‌تواند بر اساس منابع در دسترس و نوع داده‌ها انتخاب شود. هر کدام از پردازنده‌های قسمت نگاشت به صورت موازی اجرا می‌شوند.
- 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 پردازنده داده شوند. هر پردازنده کاهش، بر روی مجموعه‌ی 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 مصرف برق، گاز، و آب ساختمان‌های یک کارخانه را محاسبه کنید که در بهینه‌سازی مصرف منابع و آسیب کمتر به محیط زیست به ما کمک می‌کند. مجموعه داده‌ای از مصرف هر یک از این منابع را برای ساختمانهای مختلف در دو ماه اخیر و در ساعت‌های مختلف به شما داده می‌شود، همچنین از شما خواسته می‌شود که

³ Functional

ساعت اوج مصرف را برای هر منبع و ساختمان های مختلف و همچنین میانگین مصرف هر منبع را برای هر ساختمان محاسبه کنید.

نحوه پیاده سازی

هر ساختمان یک فولدر متناظر که نام آن، نام ساختمان می باشد دارد که شامل سه فایل CSV برای هر منبع می باشد. هر کدام از این فایل ها دارای اطلاعات مصرف یکی از منابع است که اسم منبع، اسم فایل است.

یک نمونه از فرمت این فایل ها:

Year	Month	Day	0	1	2	3	4	5
2023	01	01	12	23	24	54	8	6
2023	01	02	4	3	45	3	65	34
2023	01	03	19	23	4	7	4	67
2023	01	04	9	5	75	65	8	6

نحوه پیاده سازی به این شکل است که در ابتدا یک پردازه اولیه وجود دارد که آدرس پوشه buildings در آرگومان ورودی به آن داده می شود. این پردازه به ازای هر ساختمان، یک پردازه جدید ایجاد می کند.

هر پردازهی ساختمان، پردازهی فرزند پردازهی اولیه می باشد و یک پردازهی نگاشت برای پردازهی اصلی به حساب می آید و با خواندن و پردازش اطلاعات مربوطه، نتیجهی محاسبات را برای پردازهی پردازهی اصلی (در نقش کاهش) ارسال می کند.

پردازهی دیگری نیز ایجاد می‌شود که وظایف اداره قبوض را انجام خواهد داد (یک پردازهی قبوض برای تمام ساختمان‌ها). بدین صورت که فرزند پردازهی اولیه خواهد بود و با دریافت ساعات مصرفی از پردازهی ساختمان و محاسبه قبض هر منبع، نتیجه محاسبات را برای پردازهی ساختمان ارسال می‌کند تا ساختمان بتواند عملیات نگاشت را تکمیل کند.

هر ساختمان نیز برای سه منبع آب، برق، گاز که اطلاعات هرکدام در فایل های مجزا آورده شده پردازه مجزا ایجاد خواهد کرد.

پردازه‌های مربوط به منابع هر ساختمان نیز نقش نگاشت را ایفا خواهند کرد و با بررسی ساعات مصرفی منبع مربوطه، نتیجه را برای پردازهی کاهش (در اینجا ساختمان) ارسال خواهند کرد.

نحوه محاسبه قبض برای هر منبع بدین صورت خواهد بود:

نکات	نحوه محاسبه قبض	نام منبع
برای ساعات اوج مصرف هزینه ۱.۲۵ برابر محاسبه میشود.	مجموع آب مصرفی برحسب لیتر × ضریب مربوطه.	آب
برای ساعات اوج مصرف هزینه ۱.۲۵ برابر محاسبه میشود. در صورت مصرف کمتر از میانگین مشخص شده، ۰.۷۵ برابر میشود (پاداش مصرف بهینه)	مجموع آب مصرفی برحسب لیتر × ضریب مربوطه.	برق
-	مجموع گاز مصرفی برحسب لیتر × ضریب مربوطه.	گاز

توجه داشته باشید که برنامه شما به صورت تعاملی⁴ است؛ به صورتی که اینکه گزارش نهایی برای کدام منبع، کدام ساختمان و چه شاخصی باشد، توسط کاربر مشخص می‌شود. فرمت این دستور ها به عهده دانشجویان می‌باشد.

پردازه اولیه شما باید در ابتدای اجرا، به عنوان آرگومان فراخوانی، مسیر فولدر buildings را دریافت کند، تمام ساختمان های موجود را شناسایی کند، نام و تعداد ساختمان ها را لاگ کرده، کاربر نمایش دهد و پردازه های مدنظر را ایجاد کند. کاربر از بین منابع موجود، یک یا چند منبع را به صورت space-separated در ترمینال⁵ برنامه وارد می‌کند و در خط بعد نام یا شناسه ساختمان های(وابسته به پیاده سازی دانشجو) مدنظر را وارد میکند. سپس اطلاعات را به پردازه های فرزند مربوط به ساختمان ارسال میکند تا گزارش مربوطه را ایجاد کنند.

پس از اتمام پردازش پردازه‌ها و چاپ گزارش، برنامه پایان می‌یابد. دقت کنید که چاپ گزارش و اتمام پردازه‌های فرزند حتما قبل اتمام پردازه اولیه و برنامه اصلی انجام شود. همچنین تضمین می‌شود کاربر ورودی معتبر و با فرمت درست وارد خواهد کرد. شاخص‌هایی که در این پروژه باید به دست آورده شوند، شامل میانگین مصرف هر منبع در ماه، مصرف کل آن منبع در هر ماه، ساعات پرمصرف آن منبع در هر ماه و قبض مصرفی هر منبع در هر ماه و اختلاف مصرف در ساعات اوج با میانگین مصرف می‌باشد.

دقت کنید که برای انتقال اطلاعات لازم بین هر دو پردازه پدر و فرزند باید از unnamed pipe استفاده شود. برای انتقال اطلاعات مورد نیاز بین دو پردازه که ارتباط پدر-فرزندی ندارند، از named pipe استفاده می‌شود.

⁴ interactive

⁵ terminal

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.⁶
- 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.
- هر برای هر گونه ارتباط و پیام‌رسانی بین پردازنده‌ها باید `log` مناسب در مکان مناسبی چاپ شود تا روند اجرای برنامه قابل بررسی و صحت‌سنجی باشد.
- تمیزی کد شما نمره امتیازی دارد.
- تصور شود هر سال ۱۲ ماه و هر ماه ۳۰ روز دارد.

⁶ `close()`

- رابط کاربری مناسب و قابل درک برنامه شما (به گونه ای که کسی که کد شما را ندیده هم بتواند آنرا اجرا کند و نتیجه بگیرد) بخشی از نمره پروژه را تشکیل می‌دهد و حتما به آن دقت شود.
- ظاهر زیبای ترمینال برای شما نمره امتیازی به همراه خواهد داشت.

نکات پایانی

- برنامه حتما باید با استفاده از makefile و کامپایلر ++g اجرا شود و به راحتی فایل های لازمه ایجاد شوند.
- برنامه باید در سیستم عامل لینوکس و در زمان معقول اجرا شود.
- 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.

➤ pouyasadeghi2012@gmail.com

➤ sadeqiansoroosh@gmail.com