

پروژه درس سیستم عامل

دانشکده مهندسی برق و کامپیوتر

مهلت تحویل:

طراح پروژه:

استاد: دکتر محمدرضا حیدریور

۱۴۰۳/۱۰/۲۵ ساعت ۲۳:۵۹

علیرضا پیمانی راد

مقدمه

در این پروژه می خواهیم 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 داده‌هایی از ژانر کتب 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 سخت افزاری را داشتند اما امروزه برای بسیاری از موارد این امر غیر ممکن است. نگاشت کاهش یک چارچوب و مدل برنامه نویسی است که اجازه اجرای 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 تابع‌گرا^۳ مانند Lisp گرفته شده است. در ادامه به توضیح این مدل خواهیم پرداخت:

مدل نگاشت کاهش 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.

^۱ process

^۲ Map-Reduce

^۳ Functional

شرح پروژه

در این پروژه شما قرار است به روش نگاشت-کاهش، عمل شمارش تعداد کتاب‌ها را بر اساس ژانر انجام دهید. مجموعه داده‌ای از کتاب‌های یک کتابخانه و ژانرهای آن‌ها در اختیار داریم و می‌خواهیم بدانیم در هر ژانر، چه تعداد کتابی در این کتابخانه وجود دارد.

نحوه پیاده سازی

یک پوشه به نام `library` به شما داده می‌شود که حاوی تعدادی فایل `csv` با نام `part#num.csv` می‌باشد. در هر کدام از این فایل‌ها اطلاعات کتاب‌های هر بخش از کتابخانه وجود دارد. به این صورت که در هر خط نام یک کتاب و ژانرهای آن کتاب نشان داده شده است. برای مثال، `part1.csv` می‌تواند به شکل زیر باشد:

Invisible Man, Novel

The Guns of August, Historical

Bird Box, Horror

همچنین در پوشه `library` یک فایل `csv` دیگر به نام `genres.csv` وجود دارد که شامل یک خط است و تمام ژانرهای موجود در کتابخانه در آن لیست شده است.

نحوه پیاده‌سازی به این شکل است که در ابتدا یک پردازه اولیه وجود دارد که به آن آدرس پوشه `library` داده و به ازای هر فایل `CSV` مربوط به بخش‌ها، یک پردازه جدید ایجاد می‌کند. حال هر پردازه مربوط به یک بخش، با خواندن `CSV` متناظر و پردازش داده‌ها، تعداد کتاب‌های مربوط به هر ژانر در آن بخش را محاسبه می‌کند و به پردازه متناظر آن ژانر که قرار است تعداد کل کتاب‌های کتابخانه با آن ژانر را محاسبه کند، انتقال می‌دهد. همانطور که در بالا اشاره شد، به ازای هر ژانر یک پردازه جداگانه خواهیم داشت که توسط پردازه اولیه ایجاد می‌شود و تعداد کل کتاب‌های کتابخانه با آن ژانر را توسط اطلاعاتی که از پردازه‌های بخش‌های مختلف می‌گیرد، محاسبه می‌کند. توجه داشته باشید که نام و تعداد کل ژانرها در ابتدا نامعلوم است و باید با خواندن فایل `genres.csv` و در پردازه اولیه، این اطلاعات را به دست آورید.

دقت کنید که برای انتقال اطلاعات لازم بین پردازه پدر و فرزند (مثل انتقال نام ژانرها در صورت نیاز به فرزندان) باید از `named pipe` استفاده می‌شود. برای انتقال تعداد کتاب‌های یک ژانر در یک بخش به پردازه متناظر آن ژانر از `named pipe` استفاده می‌شود (مثلاً برای انتقال تعداد کتاب‌های ژانر `Horror` هر بخش، حتماً باید یک `named pipe` جداگانه بین پردازه مربوط به آن بخش و پردازه متناظر با ژانر `Horror` ساخته شود).

ورودی و خروجی برنامه

برای اجرای برنامه خود از دستور زیر استفاده کنید:

./GenreCounter.out library

خروجی برنامه شما نیز باید در هر خط نام ژانر و تعداد کتاب های مربوط به آن ژانر را چاپ کند. ترتیب ژانرها اهمیت ندارد.

نکات تکمیلی

- همراه با توضیحات پروژه، تعدادی فایل برای شما آپلود شده است. تنها کافی است بخش های مشخص شده ای از این فایل ها را تکمیل کنید. (بخش هایی که باید تکمیل شوند، با کامنت توضیح داده شده اند و مشخص است چه کاری باید انجام شود).
- برای ساخت و اجرای پروژه ها حتماً از فراخوانی های سیستمی `fork` و `exec` استفاده کنید. (استفاده از روش های دیگر برای ایجاد پروژه ها مجاز نیست).
- پس از استفاده از `pipe` ها، حتماً آن ها را ببندید تا از مشکلات مربوط به منابع جلوگیری شود.
- به دلیل استفاده از نوع `pipe` ها در هر مرحله فکر کنید. در زمان تحویل سؤالاتی در این باره پرسیده خواهد شد.
- دقت شود تنها راه ارتباطی میان پروژه ها استفاده از `pipe` است و هیچ راه دیگری قابل قبول نیست.
- هیچ نوع دیگری از پیاده سازی بجز مدلی که در بالا توضیح داده شد قابل قبول نیست.

نکات پایانی

- برای کامپایل برنامه خود، تنها از `makefile` ارائه شده استفاده کنید. (برنامه هایی که بدون استفاده از این `makefile` کامپایل شوند، بررسی نخواهند شد).
- برای خواندن فایل ها و عملیات مشابه، فقط مجاز به استفاده از توابع تعریف شده در فایل `utils.cpp` هستید.
- تمامی نتایج را در یک فایل فشرده با نام `OS-Project-<N_S>.zip` در سامانه یکتا آپلود کنید. (<N_S> نام و شماره دانشجویی شما است).
- انجام این پروژه به صورت انفرادی است.
- در صورت داشتن هرگونه سوال با این پروژه فقط با آیدی [@AlirezaPRad](#) در ارتباط باشید.

شاد باشید: