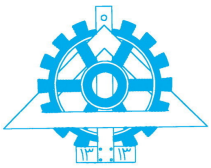
پروژه دوم درس



دانشکده مهندسی برق و کامپیوتر | **سیستم‌های عامل - پاییز ۱۴۰۰**

استاد: | مهلت تحویل: | **طراحان:**

دکتر مهدی کارگهی | یک‌شنبه ۳۰ آبان | **امید بداقی، علیرضا سلامت**

مقدمه

هدف از انجام این پروژه آشنایی با نحوه مدیریت کردن پردازه[1]ها و راه‌های ارتباطی میان آنها است. در این پروژه به پیاده‌سازی یک شمارنده کلمات با استفاده از چارچوب Map-Reduce پرداخته می‌شود.

Map-Reduce

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. Map-Reduce یک چارچوب و مدل برنامه‌نویسی برای پردازش داده‌های حجیم است و بسیاری از مفاهیم آن از زبان‌های تابع‌گرا[2] مانند Lisp گرفته شده‌است. در صورت علاقه می‌توانید از این لینک برای آشنایی بیشتر با Map-Reduce استفاده کنید. در ادامه به توضیح این مدل خواهیم پرداخت:

---

[1] Process

[2] Functional

Map-Reduce از دو بخش نگاشت[3] و کاهش[4] تشکیل می‌شود.

- در قسمت نگاشت، تعدادی پردازه برای عمل نگاشت وجود دارد که کاملا 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 پردازه‌های قسمت نگاشت محدودیت خاصی ندارد و می‌تواند بر اساس منابع در دسترس و نوع داده‌ها انتخاب شود. هر کدام از پردازه‌های 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، حتما باید به یک پردازه داده شوند. هر پردازه کاهش، بر روی مجموعه داده‌های با کلید یکسان، عملیات موردنظر را انجام می‌دهد و خروجی را ایجاد می‌کند.

در ادامه با انجام یک مثال بسیار ساده‌شده که مربوط به پروژه است، با مفاهیم به صورت کامل آشنا خواهید شد. فرض کنید یک پوشه شامل چند فایل متنی بسیار حجیم در اختیار شما گذاشته شده‌است و شما باید تعداد کلمات درون این فایل‌ها را بدست بیاورید. برای مثال، فرض کنید دو فایل A.csv و B.csv وجود دارد. محتوای آنها به صورت زیر است:

A.csv

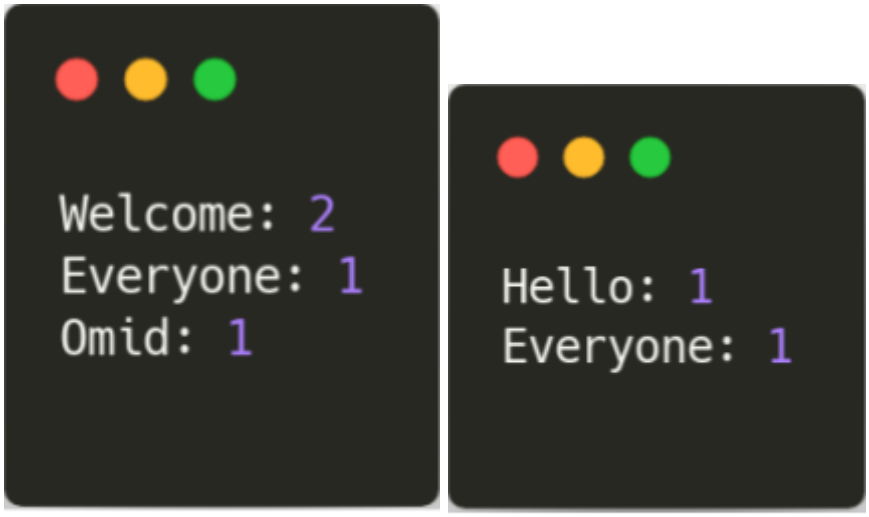
B.csv

[3] Map

[4] Reduce

در مرحله نگاشت می‌توان برای هر فایل یا حتی برای هر خط از یک فایل، یک پردازه نگاشت مجزا در نظر گرفت. حالت اول را در نظر بگیرید.

یک پردازه روی فایل اول و یک پردازه روی فایل دوم به صورت مجزا اجرا می‌شوند و خواهیم داشت:



Process1 Process2

حال در قسمت کاهش، کلیدهای یکسان به یک پردازه داده می‌شوند. البته ممکن است دو کلید متفاوت به یک پردازه داده شود اما امکان ندارد دو داده با کلید یکسان در دو پردازه مجزا قرار گرفته باشند. فرض می‌کنیم دو پردازه کاهش داریم.

کلیدهای Welcome و Everyone به پردازه اول و کلیدهای Omid و Hello به پردازه دوم داده می‌شوند. بر اساس نوع مساله که شمارش تعداد کلمات است، هر پردازه value هایی که کلید یکسان دارند را با یکدیگر جمع می‌کند و در خروجی می‌نویسد:

پس خروجی هر پردازه به صورت زیر خواهد بود:

Process1 Process2

در این مرحله کار تمام شده‌است و می‌توان خروجی را ثبت کرد.

همانطور که گفته شد، این مدل بسیار ساده شده از مدل Map-Reduce است و بسیاری از مراحل برای سادگی در نظر گرفته نشده‌اند.

شرح تمرین

در این پروژه، به پیاده‌سازی شمارنده کلمات با استفاده از روندی که در بالا توضیح داده‌شد خواهید پرداخت. دقت شود یک پوشه به نام testcases به شما داده می شود که حاوی تعدادی فایل csv می باشد. اسم فایل ها به صورت <num>.csv می باشد. هر کدام از این فایل ها حاوی تعدادی کلمه است که با , از یکدیگر جدا شده‌اند. برای مثال، فایل‌ها می‌توانند به شکل زیر باشند:

فایل 1.csv :

there,is,an,old,time,toast,which,is,golden,for,its,beauty

فایل 2.csv :

Hi,there,is,a,golden,goal,which,can,eat,toast

معماری سامانه

برای این سامانه سه نوع پردازه با نام‌های پردازه اصلی، پردازه نگاشت و پردازه کاهش در نظر گرفته شده‌است که در ادامه توضیحات آنها به صورت کامل بیان می‌شود.

1. پردازه اصلی

   این پردازه، پردازه والد سامانه محسوب می‌شود و وظیفه آن، بوجود آوردن پردازه‌های کاهش و نگاشت است. دقت شود فقط یک پردازه اصلی در سامانه وجود دارد. این پردازه، باید به تعداد فایل های csv موجود در دایرکتوری testcases، پردازه فرزند از نوع نگاشت ایجاد کند و نام فایل آنها را مشخص کند. **دقت شود برای انتقال نام فایل به پردازه نگاشت حتما از unnamed pipe[5] استفاده شود.** همچنین **یک** پردازه از نوع کاهش نیز در این پردازه ساخته می‌شود که خروجی نهایی را به پردازه اصلی ارسال می‌کند. در این مرحله پردازه اصلی باید نتایج را در فایل output.csv ثبت کند.

2. پردازه نگاشت

[5] https://www.geeksforgeeks.org/pipe-system-call/

هر کدام از پردازه‌های نگاشت نام فایلی را که باید بر روی آن عملیات انجام دهند، دریافت کرده‌اند. در این مرحله، هر پردازه تعداد رخداد کلمات در فایل مربوطه را محاسبه می‌کنند. خروجی این قسمت تعدادی کلید و مقدار متناظر آنها (یعنی کلمه‌ها و تعداد تکرار آن ها) است. هر پردازه نگاشت، خروجی خود را با استفاده از named pipe[6] به پردازه کاهش ارسال می‌کند.

3. پردازه کاهش

این پردازه که توسط پردازه اصلی ساخته شده‌است، منتظر می‌ماند تا خروجی تمام پردازه‌های نگاشت را دریافت کند. در این مرحله، خروجی‌ها با یکدیگر ترکیب می‌شوند و تعداد رخداد هر کلمه در میان تمام فایل‌ها محاسبه می‌شود. در نهایت، این خروجی با استفاده از unnamed pipe به پردازه اصلی فرستاده می‌شود.

مثال خروجی:

there: 2

is: 3

an: 1

old: 1

time: 1

toast: 2

which: 2

golden: 2

for: 1

its: 1

beauty: 1

Hi: 1

a: 1

[6] https://www.geeksforgeeks.org/named-pipe-fifo-example-c-program/

goal: 1

can: 1

- برای ساخت پردازه‌ها توسط پردازه اصلی، حتما از فراخوانی‌های سیستمی fork و exec برای ساخت و اجرای آنها باید استفاده کنید.
- ترتیب کلمه ها در خروجی اهمیتی ندارد.
- فرمت انتقال داده‌ها میان پردازه‌ها بر عهده خودتان است.
- بعد از استفاده از پایپ‌ها، آنها را ببندید.[7]
- 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 پردازه‌ها استفاده از pipe است و هیچ راه دیگری قابل قبول نیست.
- هیچ نوع دیگری از پیاده‌سازی بجز مدلی که در بالا توضیح داده‌شد قابل قبول نیست.

نکات پایانی:

- برنامه حتما باید با استفاده از makefile و کامپایلر g++ اجرا شود.
- برنامه باید در سیستم‌عامل 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 تی ای های این پروژه در ارتباط باشید.
  - alireza.salamat@ut.ac.ir
  - omid.bodaghi79@gmail.com

[7] close()