



## پروژه دوم درس



دانشکده مهندسی برق و کامپیوتر

مهلت تحویل:

سیستم های عامل - پاییز ۱۴۰۱

استاد: دکتر مهدی کارگهی

۱۴۰۱/۰۹/۰۱ ساعت ۲۳:۵۹

طراحان تمرین:

علیرضا سلامت، محمد فرهی، علی

عباسی

### مقدمه

در این پروژه می‌خواهیم با نحوه مدیریت کردن پدازه<sup>۱</sup>ها و روش‌های ارتباطی آنها آشنا شویم. در این تمرین با بهره‌گیری از عملیات‌هایی در سطح پدازه‌ها محاسباتی را روی داده‌هایی از ژانر کتب انجام خواهید داد.

### مدل نگاشت کاهش<sup>۲</sup>

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 پردازش

<sup>۱</sup> process

<sup>۲</sup> Map-Reduce

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<sup>3</sup> 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 پردازه 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 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، حتماً 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.

## 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.

---

<sup>3</sup> Functional

## نحوه پیاده سازی

یک پوشه به نام library به شما داده می شود که حاوی تعدادی فایل csv با نام part#num.csv می باشد. در هر کدام از این فایل ها اطلاعات کتاب های هر بخش از کتابخانه وجود دارد. به این صورت که در هر خط نام یک کتاب و ژانر های آن کتاب نشان داده شده است. برای مثال، part1.csv می تواند 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 ژانر های موجود در کتابخانه در آن لیست شده است.

نحوه پیاده سازی به این شکل است که در ابتدا یک پردازه اولیه وجود دارد که به آن آدرس پوشه library داده می شود و به ازای هر فایل csv مربوط به بخش ها، یک پردازه جدید ایجاد می کند. حال هر پردازه مربوط به یک بخش، با خواندن فایل csv متناظر و پردازش داده ها، تعداد کتاب های مربوط به هر ژانر در آن بخش را محاسبه می کند و به پردازه متناظر آن ژانر که قرار است تعداد کل کتاب های کتابخانه با آن ژانر را محاسبه کند، انتقال می دهد. همانطور که در بالا اشاره شد، به ازای هر ژانر یک پردازه جداگانه خواهیم داشت که توسط پردازه اولیه ایجاد می شود و تعداد کل کتاب های کتابخانه با آن ژانر را توسط اطلاعاتی که از پردازه های بخش های مختلف می گیرد،

محاسبه می کند. توجه داشته باشید که نام و تعداد کل ژانر ها در ابتدا نامعلوم است و باید با خواندن فایل genres.csv و در پرده اولیه، این اطلاعات را به دست آورید.

دقت کنید که برای انتقال اطلاعات لازم بین پرده پدر و فرزند (مثل انتقال نام ژانر ها در صورت نیاز به فرزندان) باید از unnamed pipe استفاده شود. برای انتقال تعداد کتاب های یک ژانر در یک بخش به پرده متناظر آن ژانر از named pipe استفاده می شود (یعنی مثلاً برای انتقال تعداد کتاب های ژانر Horror هر بخش، حتماً باید یک named pipe جداگانه بین پرده مربوط به آن بخش و پرده متناظر با ژانر Horror ساخته شود). در نهایت هر کدام از پرده های متناظر با ژانر ها، باید پس از اتمام شمارش، نتیجه را چاپ کنند.

## ورودی و خروجی برنامه

برنامه شما باید آدرس مربوط به پوشه library را به صورت آرگومان در رابط خط فرمان دریافت کند. همچنین نام نهایی برنامه قابل اجرا GenreCounter.out باشد. به طور مثال نمونه ورودی به شکل زیر است:

```
./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 ژانر ها اهمیت ندارد.

## نکات تکمیلی

- برای ساخت پردازها توسط پرداز اصلی، حتما از فراخوانی‌های سیستمی fork و exec برای ساخت و اجرای آنها باید استفاده کنید.
- فرمت انتقال داده‌ها میان پردازها بر عهده خودتان است.
- بعد از استفاده از پایپ‌ها، آنها را ببندید.<sup>4</sup>
- به دلیل استفاده از نوع pipe ها در هر مرحله فکر کنید. در زمان تحویل سولاتی در این باره پرسیده خواهد شد.
- دقت شود تنها راه ارتباطی میان پردازها استفاده از pipe است و هیچ راه دیگری قابل قبول نیست. اطلاعاتی که از طریق آرگومان به پردازها منتقل می شود باید بسیار ناچیز باشد.
- هیچ نوع دیگری از پیاده‌سازی بجز مدلی که در بالا توضیح داده شد قابل قبول نیست.

## نکات پایانی

- برنامه حتما باید با استفاده از makefile و کامپایلر ++g اجرا شود.
- برنامه باید در سیستم عامل لینوکس و در زمان معقول اجرا شود.
- تمامی نتایج را در یک فایل فشرده با اسم zip <#SID>-OS-CA2 در محل بارگذاری درس آپلود کنید.
- انجام این پروژه به صورت انفرادی است.

---

<sup>4</sup> close()

- 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.

➤ [mohammad.farrahi@ut.ac.ir](mailto:mohammad.farrahi@ut.ac.ir)

➤ [alireza.salamat@ut.ac.ir](mailto:alireza.salamat@ut.ac.ir)

➤ [aliabbasi806@gmail.com](mailto:aliabbasi806@gmail.com)