



به نام خدا
سیستم عامل
پروژه شماره ۲ : پردازش
زمان تحویل : ۲۰ آبان



هدف پروژه:

در این پروژه قصد داریم سیستم جست و جوی^۱ یک فروشگاه اینترنتی را پیاده سازی کنیم. این سیستم در تلاش است با خواندن چندین فایل که هر خط موجود در آن ها مشخصات یک محصول است جست و جوی مورد نظر کاربر را برگرداند.

جست و جوی مورد نظر کاربر از حداکثر دو قسمت تشکیل شده است. یک قسمت نوع فیلترینگ محصولات را نشان می دهد و قسمت دیگر نوع مرتب سازی را نشان می دهد.

در ادامه با جزییات این سیستم آشنا می شویم:

معماری سیستم:

به دلیل بالا بودن تعداد فایل های موجود برای بررسی تصمیم گرفته ایم که از چندین پردازش^۲ برای واکاوی^۳ آن ها استفاده کنیم؛ نام این پردازش ها را **worker** می نامیم.

برای تقسیم کار بین این پردازش ها نیز از یک پردازش دیگر به نام **load balancer** استفاده می کنیم که ورودی را از کاربر می گیرد و عمل واکاوی مجموعه ی فایل های انبار فروشگاه اینترنتی را بین پردازش های **worker** تقسیم می کند. تبادل داده بین پردازش **load balancer** و هر پردازش **worker** از طریق یک **unnamed pipe** است. (هر فایل از طریق یک **unnamed pipe** به یک پردازش **worker** اختصاص داده می شود).

هر پردازش **worker** پس از انتخاب محصولات موجود در فایل ها براساس قسمت فیلتر اعمال شده توسط کاربر، آن ها را برای پردازش **presenter** می فرستد و این پردازش خروجی را با توجه به قسمت مرتب سازی دستور کاربر مرتب کرده و در کنسول نمایش می دهد. تبادل داده بین سایر پردازش ها و پردازش **presenter** از طریق یک **named pipe** می باشد.

عمل مرتب سازی در پردازش **presenter** بدین ترتیب است که به ازای هر دستور این پردازش یک خروجی از هر پردازش **worker** می گیرد. این پردازش هر خروجی هر پردازش **worker** را مرتب می کند و با خروجی مرتب

¹ Search

² Process

³ Retrieve

شده‌ای که تا الان دارد ادغام^۱ می‌کند و در نهایت همین خروجی مرتب‌شده‌ی نهایی را برای کاربر نمایش می‌دهد.

* در نظر داشته باشید که در حقیقت پردازهی load balancer پدر سایر انواع پردازها است و آن‌ها را fork می‌کند و قسمت فیلترینگ و قسمت مرتب‌سازی دستور کاربر را برای پردازهای مرتبط می‌فرستد. همچنین در صورتی که ورودی quit را بگیرد باید پردازش را متوقف کند در غیر این صورت مداوم ادامه خواهد داشت.

ورودی کاربر:

پردازهی load balancer یک ورودی به شکل زیر از کاربر می‌گیرد:

(<field name> = <corresponding filtering value> -)*(<sorting value = ascend/descend> -)?

prc_cnt = n - dir = <relative dataset address>

به عنوان مثال داریم:

brand = asus - price = descend - prc_cnt = 4 - dir = laptops

که نشان‌دهنده‌ی این است که کاربر می‌خواهد جستجو فقط در بین محصولات با برند asus اتفاق بیفتد و همچنین خروجی نهایی براساس قیمت به صورت نزولی مرتب شود. همچنین prc_cnt نشان‌دهندی تعداد پردازهای worker است که قرار است برای این جستجو استفاده شوند و dir نیز نشان‌دهندی فولدری است که در آن فایل‌های مربوط به این جستجو قرار دارد. توجه داشته باشید که نام فیلدهای موجود در هر فایل dir مانند یکدیگر است و اول هر فایل نیز آورده شده است.

نمونه فایل موجود در dir:

brand model ram hdd price

apple pro 16 512 30

apple pro 16 246 26

¹ Merge

- کدهایتان باید به زبان C++ و با g++ قابل کامپایل باشند.
- نمونه فایل‌های ورودی برای بررسی در سایت قرار داده شده است.
- این پروژه تک‌نفره است.
- کد کسی را به هیچ وجه کپی نکنید، حتی یک تابع!

موفق و سربلند باشید.