



تمرین کامپیوتری شماره ۲



سیستم‌های عامل - بهار ۱۳۹۹

دانشکده مهندسی برق و کامپیوتر

طراح: حسین سلطانیلو

مهلت تحویل: ۴ فروردین ۱۳۹۹

استاد: مهدی کارگهی

مقدمه

در این تمرین قصد داریم یک سامانه جست‌وجو برای یک پایگاه اطلاعاتی از بازی‌های ویدیویی طراحی کنیم. این سامانه در تلاش است با خواندن چندین پرونده^۱ که هر خط موجود در آن‌ها مشخصات یک بازی ویدیویی را نگه‌داری می‌کند، جست‌وجوی مدنظر کاربر را انجام و نمایش دهد. برای تسریع در جست‌وجو از پرده‌ها استفاده می‌کنیم و ارتباط میان آن‌ها نیز از طریق **Pipe** صورت خواهد گرفت. این جست‌وجو از حداکثر دو بخش تشکیل شده است که در ادامه با جزئیات بیشتر به آن‌ها می‌پردازیم. به‌طور مختصر یک بخش مربوط به نوع صافی^۲‌های اعمال‌شده و دیگری مربوط به نوع مرتب‌سازی داده‌ها است. در ادامه با جزئیات بیشتری از این سامانه آشنا می‌شوید.

^۱ File

^۲ Filter

معماری سامانه

به دلیل زیادبودن تعداد پرونده‌های موجود برای بررسی، جهت تسریع در جست‌وجو تصمیم گرفته‌ایم که از چندین پردازنده³ برای واکاوی داده‌ها استفاده کنیم که آن‌ها را پردازنده‌های «کارگر»⁴ نام‌گذاری می‌کنیم.

از طرفی برای تقسیم کار بین این پردازنده‌های کارگر، باید از یک پردازنده دیگر استفاده شود که آن را نیز پردازنده «متعادل‌کننده بار»⁵ می‌نامیم. وظیفه اصلی این پردازنده این است که ورودی‌ها را از کاربر دریافت کند و سپس عملیات واکاوی مجموعه پرونده‌های این پایگاه اطلاعاتی را بین پردازنده‌های کارگر تقسیم کند.

تبادل داده میان پردازنده متعادل‌کننده بار و هر پردازنده کارگر از طریق یک **Unnamed Pipe** به ازای هر پردازنده کارگر صورت می‌گیرد و در واقع هر پرونده از طریق یک **Unnamed Pipe** به یک پردازنده کارگر اختصاص داده می‌شود. همچنین پردازنده کارگر از نوع صافی‌هایی که قرار است اعمال کند باید مطلع شود که آن‌ها نیز باید از این طریق انتقال پیدا کنند.

هر پردازنده کارگر پس از انتخاب و جداسازی بازی‌های ویدیویی بر اساس صافی‌های اعمال‌شده به‌دست کاربر، آن‌ها را برای پردازنده دیگری با نام پردازنده «ارائه‌کننده»⁶ می‌فرستد. در مرحله بعد این پردازنده ارائه‌کننده با جمع‌آوری تمامی داده‌های دریافت‌شده از پردازنده‌های کارگر، مرتب‌سازی‌هایی که کاربر درخواست کرده است را روی داده‌ها اعمال می‌کند و سپس آن‌ها را در خط فرمان نمایش می‌دهد. دقت کنید که تبادل داده‌ها میان دیگر پردازنده‌ها (کارگر و متعادل‌کننده بار) و پردازنده ارائه‌کننده نیز از طریق فقط یک **Named Pipe** در کل سامانه صورت می‌گیرد.

عمل مرتب‌سازی در پردازنده ارائه‌کننده به این ترتیب است که به ازای هر دستور کاربر، این پردازنده یک خروجی از هر پردازنده کارگر می‌گیرد، سپس این خروجی را مرتب می‌کند، با خروجی مرتب‌شده‌ای که تا الان دارد ادغام می‌کند و نهایتاً همین خروجی مرتب‌شده نهایی را به کاربر نمایش می‌دهد.

از دیگر نکاتی که باید در نظر داشته باشید این است که پردازنده متعادل‌کننده بار، پدر سایر پردازنده‌هاست که آن‌ها را با استفاده از فراخوانی سیستمی **fork** ایجاد می‌کند و بخش مربوط به صافی‌ها و نوع مرتب‌سازی را به پردازنده‌های مرتبط می‌فرستد. همچنین در صورتی که کاربر دستور **quit** را در خط فرمان وارد کند، این سامانه متوقف خواهد شد و در غیر این صورت باید دائماً به تمامی دستوراتی که کاربر وارد می‌کند پاسخ دهد.

³ Process

⁴ Worker

⁵ Load Balancer

⁶ Presenter

ورودی و خروجی برنامه

پردازه متعادل کننده بار یک ورودی به شکل زیر را از کاربر دریافت می کند:

```
((<field-name> = <query-value> - )*(<field-to-sort-by> = ascending|descending - )?
processes = <number-of-processes> - dir = <relative-dataset-address>
```

برای مثال ورودی زیر را در نظر بگیرید:

```
Platform = PS4 - Genre = Racing - NA_Sales = descending - processes = 5 -
dir = sales
```

این دستور نشان دهنده این است که کاربر قصد دارد جست و جو در میان بازی هایی انجام شود که مربوط به پلتفرم PS4 و در سبک مسابقه ای هستند انجام شود و همچنین خروجی نهایی بر حسب میزان فروش آن ها در آمریکای شمالی به شکل نزولی مرتب شود. همچنین **processes** نشان دهنده تعداد پدازه های کارگر است که قرار است برای این جست و جو استفاده شوند. نهایتاً **dir** نیز نشان دهنده پوشه ای است که پرونده های مربوط به داده های متناظر با این جست و جو در آن قرار دارد.

نمونه ای از یک فایل موجود در **dir**:

```
Name - Platform - Year - Genre - Publisher - NA_Sales - EU_Sales - JP_Sales - Other_Sales - Global_Sales
Wii Sports - Wii - 2006 - Sports - Nintendo - 41.49 - 29.02 - 3.77 - 8.46 - 82.74
Final Fantasy X - PS2 - 2001 - Role-Playing - Sony Computer Entertainment - 2.91 - 2.07 - 2.73 - 0.33 - 8.05
Minecraft - PS3 - 2014 - Misc - Sony Computer Entertainment - 1.97 - 2.51 - 0 - 0.94 - 5.42
```

نکات تکمیلی

- دقت کنید که نام ستون‌های موجود در همه پرونده‌های موجود در **dir** مانند یک‌دیگر است و نیز در ابتدای هر پرونده به‌ترتیب نوشته شده است.
- برای سادگی می‌توانید برای هر دستور واردشده از طرف کاربر، یک مجموعه پردازش جدید ایجاد کرده و منتظر خاتمه‌ی کار آن‌ها باشید و به ازای هر دستور یک مجموعه پردازش جدید ایجاد کنید.
- تأکید می‌شود که هدف از این تمرین، طراحی و استفاده صحیح از مفاهیم موازی‌سازی پردازش‌ها است و سایر پیاده‌سازی‌ها قابل قبول نیستند.
- دقت کنید که در صورتی که علاوه بر موارد ذکرشده در شرح تمرین، نیاز به ارسال اطلاعات بیش‌تری میان پردازش‌ها بود، ارتباط میان پردازش‌ها فقط از طریق **Pipe** صورت می‌گیرد و روش‌های دیگر ارتباط میان پردازش‌ها قابل قبول نیست.

نحوه تحویل

- در این پروژه به زبان **C++** (می‌توانید از نسخه ۱۱ این زبان استفاده کنید) کد بنویسید و کدهایتان با **g++** قابل کامپایل کردن باشد.
- نکته‌هایی که در جلسه توجیهی یا فروم مطرح می‌شوند مهم هستند. پس توصیه می‌شود حتماً در این جلسه شرکت کنید.
- هدف این تمرین، یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق با قوانین درس با آن برخورد خواهد شد.
- برای تحویل پروژه ابتدا یک مخزن خصوصی در سایت [GitLab](#) ایجاد کنید و پروژه خود را در آن **push** کنید. (برای عبور از تحریم می‌توانید از سرویس رایگان [شکن](#) استفاده کنید.) سپس حساب **UT_OS_TA** را با دسترسی **Maintainer** به مخزن خود اضافه کنید. نهایتاً کافیست در محل بارگذاری پروژه در سایت **CECM**، آدرس مخزن و شناسه آخرین **commit** خود را ثبت کنید.

شاد و سلامت باشید