



باسمه تعالی
سیستم عامل
پروژه ۲: پردازها
زمان تحویل: ۵ فروردین ۱۳۹۷



آشنایی با هدف کلی پروژه

در این پروژه قصد داریم سیستمی طراحی و پیاده‌سازی کنیم که متون موجود در پایگاه داده‌ی دانشگاه‌های کشور را براساس تعدادی کلیدواژه رتبه‌بندی کند. با توجه به نوع طراحی پایگاه داده (که در ادامه توضیح داده خواهد شد) ما نیاز داریم با هر بار درخواست تمامی متن‌های موجود را یک بار بررسی کرده و تعداد رخداد کلیدواژه‌های مورد نظر را در آن بیابیم. با توجه به بدیهی بودن این مسئله که تعداد این متن‌ها بسیار زیاد است تصمیم گرفتیم که این سیستم از چند پردازهی مختلف تشکیل شده باشد و جستجو در فایل‌ها مختلف را به پردازه‌های متفاوت بسپاریم تا بار کاری کمتری به هر یک از آن‌ها داده‌شده و زمان بیشتری از پردازنده به آن‌ها اختصاص یابد.

در پیاده‌سازی این پروژه شما مجدداً از معماری کلاینت و سرور که در پروژه‌ی قبل با آن آشنا شدید استفاده خواهید کرد و کدهای جداگانه‌ای برای کلاینت و سرور تحویل خواهید داد. سرور شما سیستم رتبه‌بندی متن و کلاینت‌های شما افرادی هستند که از این سیستم استفاده میکنند.

۲. ساختار کلاینت‌ها

کلاینت‌های این سیستم ساختار بسیار ساده‌ای خواهند داشت. کلاینت‌های ما با داشتن پورت و آدرس ip سرور از طریق سوکت به آن متصل شده و سپس درخواست خود را برای آن می‌فرستند. کلاینت‌ها در این سیستم این امکان را دارند که کلیدواژه‌هایی که مایل هستند متن‌های موجود در پایگاه داده براساس آن‌ها رتبه‌بندی شوند را در سه اولویت مختلف برای سرور بفرستند. واژه‌های با اولویت ۱ در صورت پیدا شدن در متن، ارزشی دو برابر واژه‌های با اولویت ۲ و ارزشی سه برابر واژه‌ها با اولویت ۳ دارند. همچنین لازم است کلاینت‌ها فیلدی با عنوان نام کاربری جهت ثبت log (که در ادامه بیشتر توضیح داده خواهد شد) بفرستند. کلاینت‌ها پس از فرستادن درخواست به صورت بلاکینگ منتظر جواب خود از سمت سرور می‌مانند.

کلاینت‌های برنامه‌ی شما باید با دستوری با فرمت زیر شروع به کار کنند:

./client - - [server ip] - - [server port] - - [username] - - [first word with priority ۱, second word with priority ۱, ...] - - [first word with priority ۲, second word with priority ۲, ...] , - - [first word with priority ۳, second word with priority ۳, ...]

۳. ساختار سرور:

۳.۱. نحوه‌ی نگهداری داده‌ها:

در توضیح سرور این سیستم، ابتدا نحوه‌ی نگهداری داده‌ها را توضیح می‌دهیم. داده‌های مورد نظر ما متن‌های مختلفی شامل پایان‌نامه‌ها، مقاله‌ها، اطلاعات دانشجویان دانشگاه‌ها هستند که به صورت فایل‌های متنی درون سیستم ما نگهداری می‌شوند. ساختار کلی داده‌ها به صورت تعدادی folder تو در تو است. سرور در هنگام اجرا شدن، آدرس یک پوشه را به عنوان آرگومان می‌گیرد که از طریق آن می‌تواند به تمامی متون مربوط به دانشگاه دست پیدا کند. پوشه‌ی موجود در بالاترین لایه نشان‌دهنده‌ی ساختمان داده‌ی ما بوده و پوشه‌های موجود در لایه‌ی پایینی آن، هریک مربوط به یکی از دانشگاه‌های کشور می‌باشند. در لایه‌های پایینی‌تر پوشه‌ها بیانگر دسته‌هایی خواهند بود که در هر لایه بر اساس معیارهایی مانند دانشکده، سال انتشار یا موضوع از هم تفکیک شده‌اند. در پایین‌ترین لایه فایل‌های متون مورد نظر ما قرار دارند. دقت کنید که عمق شاخه‌های درخت پوشه‌ها ممکن است با یکدیگر متفاوت باشد. (یک نمونه از فایل داده‌ها در کنار پروژه آپلود شده‌است.)

۳.۲. نحوه‌ی عملکرد سرور:

سرور سیستم ما از زمان بالا آمدن همواره منتظر دریافت درخواستی از سمت کلاینت‌ها است تا زمانی که از طریق دستور quit به کار آن خاتمه داده شود. این سیستم از چند نوع پردازشی مختلف تشکیل شده است که وظایف متفاوتی دارند که در ادامه به توضیح هریک از آن‌ها می‌پردازیم:

۱. **پردازشی اصلی:** این پردازش همان پردازشی اولیه است که هنگام بالا آوردن سرور ساخته می‌شود که وظیفه‌ی اصلی آن گرفتن درخواست‌ها از کلاینت‌ها است.

۲. **پردازشی مخصوص ثبت log:** در طراحی این سیستم می‌خواهیم قادر باشیم که به ازای هر درخواستی که به سیستم می‌آید یک log ثبت کنیم. به این منظور یک پردازش هنگام بالا آمدن سیستم ایجاد کرده‌ایم که مسئول انجام این کار است. این پردازش در ابتدا یک فایل به نام logs.txt در کنار سرور می‌سازد. حال به یک named pipe احتیاج داریم. پردازشی اصلی هر درخواستی که از سمت کلاینت‌ها دریافت می‌کند را ابتدا روی این named pipe می‌نویسد و سپس به رسیدگی به این درخواست می‌پردازد. پردازشی که مسئول ثبت logها است به ازای هر ۵ درخواستی که پردازشی اصلی روی named pipe می‌نویسد به صورت batch اطلاعات مربوط به آن‌ها را شامل نام کاربری و تعداد کلیدواژه‌های مورد جستجو را در logs.txt می‌نویسد.

۳. **پردازش‌های worker:** در سیستم ما تعدادی پردازش تحت عنوان worker وجود دارند که وظیفه‌ی رسیدگی به درخواست‌های کاربران را دارند. تعداد این پردازش‌ها در سیستم محدود بوده و این تعداد در هنگام بالا آمدن سرور از طریق آرگومان معلوم شده و در همان ابتدا ساخته می‌شوند. هنگامی که سرور یک درخواست دریافت کرد آن را برای رسیدگی به یکی از پردازش‌های worker که در حال حاضر در حال رسیدگی به هیچ درخواستی

نیست سپرده‌می شود. اگر در آن لحظه هیچ worker آزادی وجود نداشت درخواست موردنظر باید روی یک صف در انتظار آزاد شدن یک worker بایستد.

هنگامی که ریکوئست در اختیار یک worker قرار گرفت پردازشی worker پوشه‌ی اصلی مربوط به داده‌ها را باز کرده سپس به ازای هر یک از پوشه‌های موجود در آن یک پردازشی جدید می‌سازد. هر پردازش پوشه‌ی مربوط به خود را بررسی کرده و به ازای هر پوشه‌ی موجود در لایه‌ی جدید یک پردازش می‌سازد. این فرایند تا به جایی تکرار می‌شود که در لایه‌ی آخر هر شاخه از درخت به تعدادی فایل متنی برسیم. در این جا به ازای هر فایل متنی نیز یک پردازشی جدید می‌سازیم که مسئول جستجو در آن فایل متنی است.

هنگامی که پردازش جستجو در فایل مربوط به خود را به پایان رساند ارزشی که برای آن فایل مطابق با درخواست کاربر به دست آورده بود به پردازشی والد خود برمی‌گرداند. در لایه‌های میانی هر پردازش وظیفه دارد پاسخ‌هایی که از فرزندان خود دریافت می‌کند را با هم ادغام کرده و به صورت رتبه‌بندی شده به پردازشی والد خود بازگرداند.

در بالاترین لایه هنگامی که پاسخ نهایی به دست پردازشی worker رسید این پردازش باید پاسخ را که شامل لیستی از نام مقاله‌ها است که براساس تعداد تکرار کلیدواژه‌ها ارزش گذاری شده‌اند به کلاینت مورد نظر بفرستد. دقت کنید تمام ارتباط‌های بین یک پردازش و والد آن باید از طریق pipe اتفاق بیفتد.

نکات پروژه:

۱. در انجام این پروژه به terminate شدن به موقع پردازش‌ها بسیار دقت کنید. خاتمه‌ی به موقع پردازش‌ها بخش مهمی از نمره‌ی شما را تشکیل می‌دهند.
۲. سرور ما باید با گرفتن دستور quit به کار خود خاتمه دهد. دقت کنید که تمام پردازش‌های سیستم باید در این زمان terminate شوند.

سایر نکات:

۱. کدهای شما باید به زبان C++ بوده و با g++ قابل کامپایل کردن باشد.
۲. نمونه‌ی داده‌ی ورودی برای بررسی در سایت درس قرار داده شده‌است.
۳. حتما در جلسهی توجیهی حضور داشته باشید. نکاتی که در کلاس و فروم مطرح میشوند جزو پروژه هستند.
۴. این پروژه تک‌نفره است.
۵. کد خود را در یک فایل zip با شماره دانشجویی خود آپلود کنید.
۶. کد کسی را کپی نکنید. حتی یک تابع!