



باسمه تعالی
سیستم‌های عامل
پروژه‌ی شماره‌ی ۲ : پروژه
تاریخ تحویل: ۲۱ آبان ۱۳۹۶



۱. آشنایی با هدف کلی پروژه:

در این پروژه قرار است شما یک سیستم برای استعلام مقدار جریمه‌ی ثبت شده برای ماشین‌های موجود در سطح کشور طراحی و پیاده‌سازی کنید. این سیستم با گرفتن یک شماره ماشین (که برای هر ماشین یکتا است) مجموع هزینه‌ی قابل پرداخت برای آن خودرو به عنوان جریمه را اعلام می‌کند. تمام تخلف‌ها توسط دوربین‌های موجود در سطح کشور ثبت شده و تخلف‌های ثبت شده توسط هر دوربین درون یک فایل متنی ذخیره شده است که هر خط آن مربوط به یک تخلف بوده و شامل شناسه‌ی خودروی متخلف، تاریخ انجام تخلف و مبلغ جریمه‌ی ثبت شده برای آن تخلف می‌باشد.

در پیاده‌سازی این پروژه شما مجدداً از معماری کلاینت و سرور که در پروژه‌ی قبل با آن آشنا شدید استفاده خواهید کرد و یک کد برای کلاینت و یک کد برای سرور تحویل می‌دهید. سرور شما سیستم استعلام خلافی و کلاینت‌های شما افرادی هستند که از این سیستم استفاده می‌کنند.

با توجه به این که تعداد دوربین‌های موجود در سطح کشور بسیار زیاد است و برای پیدا کردن خلافی یک ماشین ما احتیاج به جست‌وجو در فایل‌های مربوط به تمامی این دوربین‌ها داریم تصمیم گرفتیم تا سیستم از چند پردازه‌ی متفاوت تشکیل شده باشد و جست‌وجو در هر فایل را به یکی از این پردازه‌ها بسپاریم تا زمان بیشتری از CPU به آن‌ها اختصاص یابد.

۲. ساختار کلاینت‌ها:

کلاینت‌های ما در این برنامه ساختار بسیار ساده‌ای دارند. این کلاینت‌ها با دانستن port و ip سرور از طریق socket به آن متصل شده و سپس به عنوان پیغام یک شناسه‌ی ۵ رقمی (که برای هر خودرو یکتا است) را برای سرور ارسال می‌کنند، سپس به صورت blocking منتظر پاسخ که مقدار جریمه‌ی کلی خود از طرف سرور است می‌مانند.

۳. ساختار سرور:

۳.۱. نحوه‌ی ذخیره‌ی داده‌ها:

کد مربوط به سرور در هنگام اجرا شدن آدرس یک folder را به عنوان آرگومان ورودی می‌گیرد که از طریق آن قرار است به اطلاعات ثبت شده در مورد خلاقی خودروها دست پیدا کند.

برای نگهداری داده‌ها در این مسئله از ساختار درختی folderها استفاده کرده‌ایم، به این صورت که در بیرونی‌ترین لایه‌ی داده‌ها همان پوشه‌ی ورودی قرار دارد که مربوط به اطلاعات ثبت‌شده برای کل کشور است، در داخل این پوشه تعداد دیگری پوشه قرار دارد که هر یک نشانگر یک استان بوده و لایه‌های پایین‌تر می‌توانند نشان‌دهنده‌ی مفاهیم کوچکتری مانند شهرها و مناطق مختلف موجود در یک شهر باشند. و در نهایت در داخلی‌ترین لایه به تعدادی فایل متنی می‌رسیم که همان‌طور که پیش‌تر گفته‌شد شامل اطلاعات ثبت‌شده توسط یک دوربین هستند. (لازم به ذکر است که عمق تمام شاخه‌های یک درخت لزوماً به یک اندازه نیست، مثلاً بعضی از شهرهای کوچک ممکن است منطبقه‌بندی نشده باشند!)

۳.۲. نحوه‌ی عملکرد سرور:

سرور از زمان شروع به کار تا زمانی که ما با دستور quit به کار آن خاتمه دهیم همواره روی یک port و ip خاص منتظر گرفتن درخواست است. هر بار که یک درخواست توسط سرور دریافت می‌شود سرور یک پردازشی جدید (که از این پس آن را p1 می‌نامیم) می‌سازد که مسئول اصلی رسیدگی به این درخواست است. اطلاعات لازم از طریق یک unnamed pipe در اختیار p1 قرار می‌گیرد. حال p1 با بررسی folder ای که به عنوان ورودی در اختیار برنامه قرار گرفته‌است عمل load balancing را بر عهده می‌گیرد. در هر لایه از ساختار داده‌ها به تعداد پوشه‌های موجود در آن لایه، پردازشی parent برای آن لایه fork کرده و هر پردازشی جدید مسئول رسیدگی به یکی از folderها می‌شود، تا جایی که درون این لایه‌ها آن قدر پیش برویم تا به فایل‌های متنی برسیم. در این مرحله نیز به تعداد فایل‌های متنی پردازشی جدید ایجاد می‌شود که مسئولیت جستجو در هر فایل و یافتن مجموع مبلغ جریمه‌های مربوط به خودروی مورد نظر در آن فایل بر عهده‌ی یکی از این پردازش‌هاست.

پس از بررسی هر یک از فایل‌های متنی نتیجه‌ی بررسی آن فایل به پردازشی parent آن folder داده می‌شود و به صورت بازگشتی هر پردازش یک حاصل جمع از نتایج بازگردانده‌شده توسط تمام پردازش‌های فرزندش را به پردازشی لایه بالاتر خود می‌دهد. توجه کنید که هر پردازش پس از رساندن پاسخ به لایه‌ی بالایی‌اش باید terminate شود و terminate کردن به موقع هر یک از پردازش‌های موجود در سیستم شما بخش مهمی از نمره‌ی پروژه را تشکیل می‌دهد.

در انتها زمانی که پردازهی p۱ پاسخ را از تمامی فرزندان خود دریافت کرد نتیجه‌ی حاصل از انجام جست‌وجو را روی یک named pipe قرار می‌دهد. در این‌جا نوع دیگری از پردازه وارد کار می‌شود که آن را p۲ می‌نامیم. وظیفه‌ی این نوع پردازه این است که داده‌ها را از روی named pipe موجود خوانده و آن را برای کلاینت مورد نظر که منتظر پاسخ است بفرستد. (p۲ می‌تواند از ابتدای اجرای سرور ساخته شود، فقط در هنگام پیاده‌سازی به زمان و نحوه‌ی terminate شدن این پردازه نیز دقت کنید!)

۴. سایر نکات :

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