



دانشگاه فردوسی مشهد

گروه مهندسی کامپیوتر

پروژه دوم درس سیستم عامل

عنوان

موتور جستجو به کمک نخ ها و همگام سازی

نگارش

فاطمه صوفیان

الهی متقین

استاد درس

دکتر محمد الله بخش

بهار 1400

1- مقدمه

در این تمرین برای پیاده سازی یک موتور جستجوی متن، به منظور صرفه جویی در زمان از نخ ها (threads) استفاده شده است و به کمک آن ها کلمات موردنظر را از فایل ورودی استخراج می کنیم.

اما با توجه به آن که برای نوشتن کلمات پیدا شده در فایل خروجی میان نخ ها وضعیت رقابتی به وجود می آید، برای رفع این مشکل دو روش mutex lock و semaphore را پیاده سازی می کنیم و به کمک آن ها مشکل فضای رقابتی را به کمک همگام سازی (synchronization) حل می کنیم.

2- همگام سازی

Synchronization یا سینک شدن و یا در اصطلاح فارسی همگام سازی در واقع هماهنگ کردن دقیق چندین رویداد یا منبع مختلف است. فرآیند همگام سازی راهی برای ایجاد هماهنگی میان فرآیندهایی است که از داده های مشترک استفاده می کنند. این فرایند در یک سیستم عامل در میان فرایندهای همکاری اتفاق می افتد. فرایندهای همکاری فرایندهایی هستند که منابع را به اشتراک می گذارند.

2-1- روش Semaphore

در علم رایانه نشانبر یا semaphore به متغیری گفته می شود که در محیط های همروند برای کنترل دسترسی فرایندها به منابع مشترک به کار می رود.

semaphore میتواند به صورت دودویی (که تنها دو مقدار صحیح و غلط را دارا است) یا شمارنده اعداد صحیح باشد. از semaphore برای جلوگیری از ایجاد وضعیت رقابتی میان فرایندها استفاده می گردد. به این ترتیب، اطمینان حاصل می شود که در هر لحظه تنها یک فرایند به منبع مشترک دسترسی دارد و می تواند از آن بخواند یا بنویسد.

شمارش سمافورها با دو عملیات همراه است. به طور تاریخی، به این دو عملیات V یا signal() و P یا wait() می گویند. اگر سمافوری به نام S داشته باشیم، عملیات V آن را یک واحد افزایش می دهد و عملیات P یک واحد آن را کاهش می دهد.

مقدار سمافور S نشان دهنده تعداد واحد منابعی است که در حال حاضر در دسترس هستند و پروسه ها می توانند از آن استفاده کنند. عملیات P، وقت را تلف می کند یا پروسه را به خواب می برد تا وقتی که یکی از منابع اشغال شده مجدداً آزاد شود، وقتی که منبع آزاد شد، بلافاصله به پروسه اختصاص می یابد. عملیات V برعکس است. وقتی که پروسه ای کارش را با منبع به اتمام رساند، با استفاده از عملیات V می تواند آن را آزادسازی کند تا دیگر پروسه ها بتوانند از آن استفاده کنند. خصوصیت مهم سمافور S این است که مقدار آن را تنها می توان به وسیله عملیات های V و S تغییر داد.

2-1-1- پیاده سازی semaphore در پروژه

برای استفاده از semaphore در پروژه از کتابخانه `java.util.concurrent.Semaphore` استفاده شده است. Semaphore استفاده شده باینری است و هنگامی که هر نخ در حال نوشتن در فایل خروجی است، ناحیه بحرانی را کنترل می کند. زمان اجرای برنامه با استفاده از Semaphore به طور میانگین برابر ۲۶۸۱۳۰۰ نانو ثانیه است.

2-2- mutex lock روش

یک Mutex یا «انحصار متقابل» (Mutual Exclusion) ساده ترین نوع «همگام ساز» (Synchronizer) محسوب می شود. این شیوه تضمین می کند که در هر لحظه، تنها یک نخ می تواند بخش حیاتی یک برنامه رایانه ای را اجرا کند. برای دسترسی به بخش حیاتی یک نخ باید Mutex را به دست آورد، سپس به بخش حیاتی دست می یابد و در نهایت Mutex را آزاد می کند. در این زمان همه نخ های دیگر مسدود می شوند تا این که Mutex مجدداً آزادسازی شود. به محض این که نخ از بخش حیاتی خارج شود، نخ دیگری می تواند وارد بخش حیاتی شود.

2-2-1- پیاده سازی روش mutex lock

برای استفاده از mutex lock در پروژه از کتابخانه `com.sun.corba.se.impl.orbutil.concurrent.Mutex` استفاده شده است. زمان اجرای برنامه با استفاده از mutex lock به طور میانگین برابر ۲۶۰۱۵۰۰ نانو ثانیه است.

2-3- پیاده سازی پروژه بدون نخ

در پیاده سازی پروژه بدون استفاده از نخ ها، زمان اجرای برنامه برابر با ۳۰۶۸۶۳۱۰۰ نانو ثانیه است که در مقایسه با حالتی که 4 نخ داریم، بیش از 100 برابر است.

2-4- مقایسه دو روش semaphore و mutex lock

یک شیء mutex فقط به یک موضوع اجازه می دهد تا یک منبع یا یک بخش مهم را مصرف کند، در حالی که semaphores تعداد محدودی دسترسی همزمان به یک منبع را امکان پذیر می سازد (تحت حداکثر تعداد مجاز). با استفاده از اشیاء mutex، موضوعات دیگر که می خواهند به منبع دسترسی پیدا کنند، باید در صف صبر بمانند تا اینکه موضوع فعلی با استفاده از منبع تمام شود.

در پیاده سازی که در پروژه انجام شده است، این دو روش از نظر زمانی تقریباً یکسان عمل کرده اند.

2-5- جستجو به کمک trie tree

درخت پیشوندی (trie tree) یک ساختار درخت مانند است که برای حل مسائل مرتبط با رشته‌ها (Strings) بسیار موثر است. این ساختمان داده امکان بازیابی سریع را فراهم می‌کند و اغلب برای جستجوی کلمات در دیکشنری، پیشنهاد خودکار در موتورهای جستجو و حتی مسیریابی IP یا IP routing مورد استفاده قرار می‌گیرد.

2-5-1- پیاده سازی trie tree در پروژه

به منظور افزایش سرعت از این ساختار داده استفاده شده است به این صورت که هر نخ به ازای هر خط از فایل خروجی یک درخت می‌سازد و هر یک از کلمات به عنوان کلیدهای این درخت در نظر گرفته شده است. به کمک این کار سرعت افزایش پیدا کرده است و زمان میانگین به حدود ۲۱۰۱۵۰۰ نانو ثانیه کاهش پیدا کرده است.

*** پروژه روی گیت هاب در آدرس <https://github.com/elahemotaghin/search-engin> قرار گرفته است.