باسمه تعالی

تمرین کامپیوتری شماره‌ی ۳
سیستم عامل (بهار ۹۲)
دانشکده‌ی مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
مهلت انجام ۱۳۹۲/۰۴/۱۵

در این تمرین، یک همگام‌سازی ساده بین نخ‌ها و استفاده از حافظه‌ی مشترک مورد بررسی قرار می‌گیرد. این تمرین متشکل از دو پردازه‌ی مستقل است. برنامه با اجرای پردازه‌ی نخست شروع می‌شود و یک فایل برای معرفی پارامترهای برنامه به عنوان آرگومان داده می‌شود. سپس پردازه‌ی اول، اقدام به اجرای پردازه‌ی دوم می‌نماید.

کلیات این دو پردازه این است که چند فایل متنی را از مکان مشخصی در دیسک خوانده و در جای دیگری کپی می‌نماید. در پردازه‌ی نخست، به تعداد فایل‌های ورودی، نخ ایجاد می‌شود. این نخ‌ها هر کدام شروع به خواندن محتوای فایل مربوط به خود کرده و آنرا در یک حافظه‌ی مشترک کپی می‌نماید. سپس تعدادی نخ دیگر، این داده‌ها را از روی حافظه‌ی مشترک برداشته و از طریق سوکت به پردازه‌ی دیگر ارسال می‌نمایند. در پردازه‌ی دوم، به تعداد نخ‌های فرستنده، نخ گیرنده ایجاده شده و داده‌ها را دریافت می‌نماید. داده‌های دریافت شده مجدداً در یک حافظه‌ی مشترک کپی می‌گردد و به تعداد فایل‌های مبدأ نخ‌هایی ایجاد می‌شود که هر کدام داده‌های مربوط به فایل خود را از حافظه‌ی مشترک برداشته و در فایل مقصد کپی می‌نماید.

ساختار انتقال داده

نخ‌هایی که از فایل‌های مبدأ داده را می‌خوانند، می‌بایست به صورت دوره‌ای اقدام به خواندن حجم تصادفی از فایل بنمایند. این حجم در هر دوره به صورت تصادفی به ازای هر نخ و از مقدار ۱۲۸ بایت تا مقدار یک کیلوبایت می‌باشد. نخ‌هایی که اقدام به خواندن از روی حافظه‌ی مشترک در هر دو پردازه می‌کنند نیز می‌بایست به صورت دوره‌ای اقدام به خواندن یک فقره داده از روی حافظه‌ی مشترک بنمایند. تمامی این دوره‌ها به صورت تصادفی و مستقل از هم و بین یک تا ۱۰ میکروثانیه می‌باشند و در هر دوره این مقدار تصادفی تغییر می‌کند(برای اینکار می‌توانید ریسمان را به اندازه‌ای تصادفی به خواب ببرید). پروتکل داده‌ها در سیستم شما، به خود شما بستگی دارد. لذا داده‌ها به صورت بسته‌ای و با پروتکل شما، روی حافظه‌ی مشترک قرار می‌گیرد و در شبکه ارسال می‌شود. در ضمن پیاده‌سازی‌ای که از فضای حافظه‌ی مشترک به صورت بهینه بهره ببرد، امتیاز کار شما خواهد بود.

همگام‌سازی

مدل همگام‌سازی به این صورت است که در پردازه‌ی نخست، هنگامی که یک نخ به حافظه‌ی مشترک دسترسی دارد، هیچ نخ دیگری حق دسترسی همزمان به حافظه‌ی مشترک را ندارد. دلیل اینکار، نویسنده بودن تمامی نخ‌ها در این بخش می‌باشد. امّا در پردازه‌ی دوم، نخ‌هایی که داده‌ها را از شبکه می‌گیرند و در حافظه‌ی مشترک می‌نویسند، نخ نویسنده‌اند. لذا هنگام دسترسی آنان به حافظه‌ی مشترک، هیچ نخ دیگری اجازه دسترسی به حافظه‌ی مشترک را ندارد. لیکن نخ‌هایی که اقدام به برداشتن داده‌ها از حافظه‌ی مشترک می‌کنند، در دو مقطع عمل می‌کنند. نخست زمانی که مشغول خواندن حافظه‌ی مشترک به منظور پیدا کردن بسته‌ی داده‌ی مربوط به فایل در نظر گرفته شده برای نخ هستند. در این حالت چندین نخ به صورت همزمان می‌توانند به حافظه دسترسی داشته باشند. اما هنگامی که داده در فایل نوشته شد، برای برداشتن آن از حافظه‌ی مشترک، به حالت نوشتن رفته و در این حالت فقط یک نخ می‌تواند روی حافظه‌ی مشترک بنویسد. تا وقتی‌که در فضای حافظه‌ی مشترک جای خالی وجود دارد، نخ‌های نویسنده‌ای که داده‌ی جدید درج می‌کنند، نسبت به دیگران از اولویت بالاتری برخوردار هستند. یعنی اگر همزمان چند نخ منتظر گرفتن قفل‌ها هستند، نخ‌های نویسنده اولویت بالاتری دارند. نخ‌هایی که قصد حذف داده دارند، اولویت بالاتری نسبت به کل نخ‌های موجود خواهند داشت.

کتابخانه‌ها و فراخوانی‌های سیستمی

برای حافظه‌ی مشترک از فراخوانی‌های سیستمی shmget, shmctl, shmat, schmdt استفاده کنید. برای همگام‌سازی هم می‌توانید از هدر semaphore.h از کتابخانه‌ی استاندارد GCC کمک بگیرد. برای مدیریت نخ‌ها هم از کتابخانه‌ی pthread.

نکات :

- سیگنال‌هایی که امکان رخداد آن‌ها می‌رود باید پیاده سازی ومدیریت شوند و تشخیص سیگنال‌های مورد نیاز برعهده‌ی شماست.
- استفاده از هر گونه متغیر سراسری بین نخ‌ها مجاز نمی‌باشد. مشخصات حافظه مشترک نیز باید به صورت آرگومان (مشابه دو پردازه مجزا) به نخ فرستاده شود.
- سوالات خود را در فروم درس بپرسید تا دیگران هم استفاده کنند. در ضمن به سوالات مطرح شده توسط دوستان خود –اگر پاسخش را می‌دانید- پاسخ دهید.
- با هرگونه کد مشابه برخورد جدی می‌شود.
- تمام قسمت‌ها باید توسط زبان C نوشته شود وپروژه‌ی شما فقط با gcc کامپایل می‌شود.
- Makefile بخشی از نمره‌ی شماست.
- گروه‌ها می‌بایست دو نفری باشند.
- تمام فایل‌های مربوط به پروژه را درون یک فولدر با **شماره دانشجویی** اعضای گروه ریخته و فولدر را **zip** کنید.