



دانشکده مهندسی کامپیوتر

دکتر رضا انتظاری ملکی

پاییز ۱۴۰۰

---

## تمرین عملی سری پنجم سیستم‌های عامل

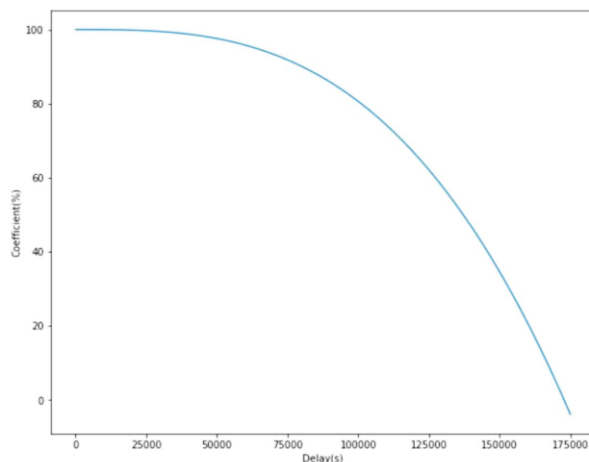
کیوان داداش‌زاده - امیر اسمعیلی

تاریخ تحویل: ۱ دی ساعت ۲۳:۵۹:۵۹

---

## قوانین

- در صورت مشاهده‌ی هرگونه تقلب، به ازای هربار تقلب نمره‌ی کل آن تمرین صفر در نظر گرفته می‌شود و هم‌چنین یک نمره (نمره منفی) از نمره‌ی کل تمرین‌ها کسر می‌شود.
- در صورت وجود هرگونه سوال از طریق گروه تلگرام مطرح کنید. (لطفا پی‌وی پیام ندهید.)
- لطفا پاسخ هر سوال را به صورت جداگانه داخل کوئرا ارسال کنید. از ارسال پاسخ‌های خود داخل تلگرام و سایر شبکه‌های اجتماعی خودداری کنید.
- در سوال تئوری (سوال چهارم)، تمامی پاسخ‌های شما باید دارای توضیحات کافی باشد. دقت کنید جواب آخر بدون توضیح هیچ نمره‌ای ندارد.
- به دلیل فشرده‌بودن برنامه‌ی ترم، امکان تمدید تمرین وجود ندارد. لطفا از روزهای ابتدایی به حل تمرین بپردازید و انجام تمرین را به روزهای پایانی موکول نکنید.
- محور افقی این نمودار، مقدار تاخیر به ثانیه و محور عمودی ضریب اعمالی در نمره تمرین است.



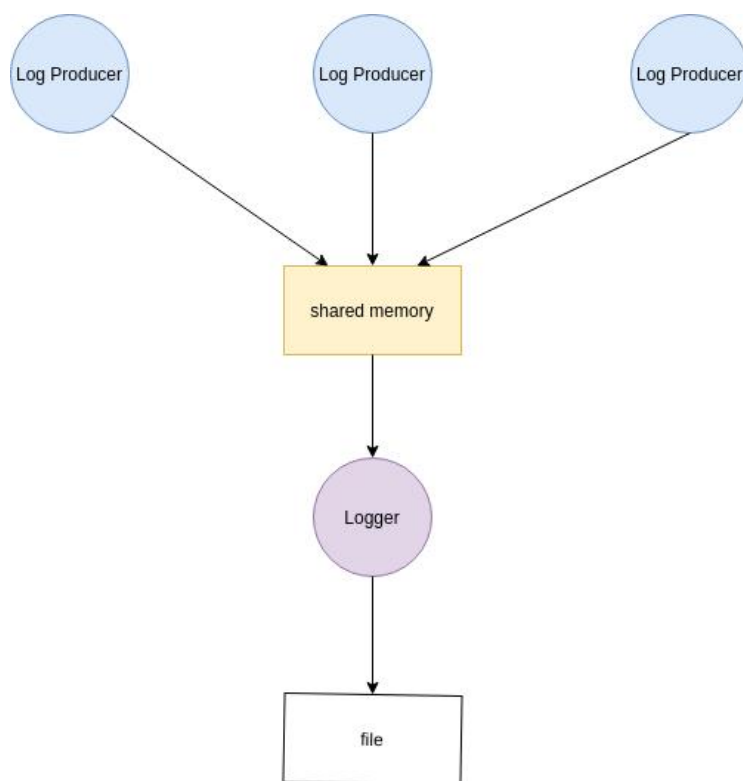
## سوالات

## ۱ Simple Logger (۳۵ نمره)

در این سوال، قصد داریم به پیاده‌سازی یک Logger ساده بپردازیم.

در ابتدای کار، شما یک پردازش را اجرا می‌کنید که وظیفه‌ی این پردازش، ثبت لاگ‌هایی است که از سمت پردازش‌های دیگر می‌آیند. (تمامی لاگ‌ها در یک فایل متنی نوشته می‌شوند.) سایر پردازش‌ها به ارسال لاگ‌های تولیدشده توسط یک مموری مشترک می‌پردازند تا پردازش تولیدکننده‌ی لاگ، آنها را داخل فایل مربوطه بنویسد.

دقت کنید در این سوال، فقط یک مکان برای اشتراک‌گذاری لاگ‌ها وجود دارد و شما مجاز به استفاده از مموری مشترک به صورت دودویی نیستید. درواقع بین پردازش‌هایی که اقدام به تولید لاگ می‌کنند و پردازش‌ای که لاگ را می‌نویسد، فقط یک فضای مشترک وجود دارد.



### نکات مهم:

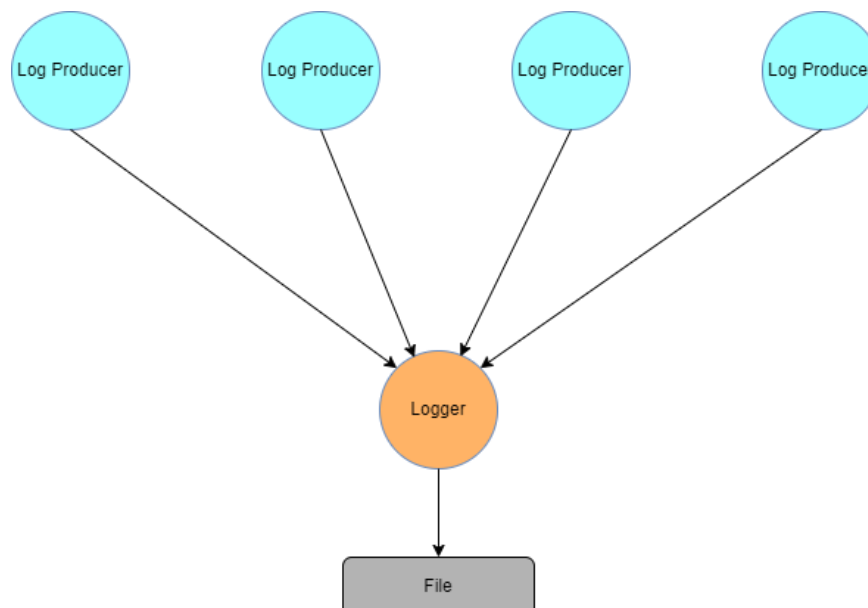
۱. برای عملیات Shared Memory در این سوال، باید از توابع System V استفاده کنید.
۲. برای همگام‌سازی پردازش‌های تولیدکننده‌ی لاگ و پردازش‌ای که به یادداشت لاگ‌ها در فایل می‌پردازد، از Semaphore استفاده کنید. (هریک از لاگ‌های مربوط به هر پردازش باید در یک خط جداگانه باشد). سعی کنید لاگ‌ها را به فرمت زیر بنویسید:

worker\_id: [log]

۳. هیچ محدودیتی برای تعداد پردازش‌های تولیدکننده‌ی لاگ وجود ندارد.
۴. برای تست بهتر و قابل درک بودن، پیشنهاد می‌شود از دو سورس کد مجزا (یکی در نقش پردازش‌ی ثبت‌کننده‌ی لاگ و دیگری در نقش تولیدکننده‌ی لاگ) استفاده کنید و آنها را جداگانه کامپایل کنید.

## ۲ Simple Logger with Pipe (۳۰ نمره)

پیاده‌سازی سوال قبل را این بار با استفاده از Pipe انجام دهید. با این تفاوت که فرض کنید بین پردازش‌های تولیدکننده‌ی لاگ و پردازش‌ی ثبت‌کننده‌ی لاگ، یک مسیر اختصاصی وجود دارد.



### ۳ Merge Sort (۲۵ نمره)

فرض کنید در یک پردازنده، یک پردازهی مادر و چند پردازهی فرزند داریم. (تعداد پردازها توانی از ۲ می‌باشد). در این سوال از شما انتظار می‌رود الگوریتم Merge Sort را به صورت توزیع شده پیاده‌سازی کنید.

حالت عملیاتی که باید انجام دهید، مرتب‌سازی یک آرایه است. به این صورت که ابتدا آرایه‌ی داده‌شده را به طور برابر بین تمام پردازه‌های فرزند تقسیم می‌کنیم. (فرض کنید آرایه‌ی ورودی به پردازهی مادر داده می‌شود). هرکدام از این آرایه‌های کوچکتر پس از مرتب‌سازی به پردازهی مادر ارسال می‌شوند. پردازهی مادر پس از دریافت تمام آرایه‌های مرتب‌شده، آنها را ادغام کرده و سپس چاپ می‌کند.

توجه کنید در حل این سوال حتما باید از روش POSIX برای عملیات Shared Memory استفاده کنید.

### ۴ Research Questions (۱۰ نمره)

با تحقیق و جستجو درمورد هریک از سوالات زیر، پاسخ مناسب را بیابید. (ذکر منبع الزامی است).

۱. دلیل وجود چندین مدل مختلف برای به اشتراک‌گذاری مموری چیست؟ هرکدام چه مزایا و معایبی دارند؟ هریک از آنها را از جهات سادگی، کارایی و دیگر موارد با یکدیگر مقایسه کنید.
۲. تحقیق کنید آیا می‌توان Mutex را بین چند پردازه به اشتراک گذاشت؟ توضیح دهید.
۳. آیا می‌توان با گرفتن فضایی از فضای Heap، آن را بین پردازه‌های مختلف به اشتراک گذاشت؟ اگر بله، توضیح دهید چرا و یک نمونه کد برای آن بیاورید. درغیراین‌صورت، با ذکر دلیل راهکار جایگزین پیشنهاد دهید.

موفق باشید.