



تکلیف چهارم سیستم عامل

دکتر زینب زالی

دانشکده مهندسی برق و کامپیوتر

تاریخ تحویل: ۹ دی ۱۴۰۲

۱. میزبان یک مهمانی، $N > 2$ نفر مهمان را به خانه دعوت کرده است. میزبان نمی خواهد چندین دفعه در خانه را برای ورود میزبانان باز کند. N مهمان برای یکدیگر صبر می کنند و به یکباره وارد می شوند. میزبان و مهمانان با برنامه چند نخ پیاده سازی می شوند.

میزبان برای ورود همه مهمانان صبر می کند سپس `openDoor()` را فراخوانی می کند و یک متغیر شرطی را سیگنال می کند. مهمانان باید برای ورود N نفر و باز شدن در صبر کنند و `enterHouse()` را فراخوانی کنند.

تنها با استفاده از متغیرهای تعریف شده در کد میزبان، کد مهمان را بنویسید.

```
//host
lock(m)
while (guest_count < N) wait(cv_host, m)
openDoor()
signal(cv_guest)
unlock(m)
```

۲. مسئله خوانندگان نویسنده را با اولویت نویسنده به خوانندگان در نظر بگیرید. می خواهیم lockهای مخصوصی با نام reader-writer locks داشته باشیم که بتوان در چنین شرایطی از آنها استفاده کرد.

سودو کدهای لازم را برای پیاده سازی توابع `readLock`، `writeLock`، `readUnlock` و `writeUnlock` بنویسید.

در پیاده سازی خود از `mutex` و `conditional variable` استفاده کنید (از سمافور استفاده نکنید).

۳. می خواهیم گذر ماشین ها از روی یک پل یک طرفه را مدیریت کنیم به صورتی که تا وقتی ماشین هایی در حال عبور از پل از یک سمت هستند به ماشین های دیگر که از سمت دیگر قصد ورود دارند اجازه ی ورود داده نشود. محدودیتی روی تعداد ماشین هایی که روی پل هستند وجود ندارد. اما می خواهیم مسئله ی گرسنگی را تا حدی حل کنیم.

بدین منظور اگر ۵ ماشین از سمت شمال از پل رد شدند و تعدادی (یک یا بیشتر) ماشین در سمت جنوب قصد ورود به پل را داشتند، از ادامه عبور ماشین ها از سمت شمال جلوگیری کرده تا ماشین هایی از سمت جنوب از پل رد شوند. همین قانون برای سمت جنوب هم به صورت برعکس برقرار است.

با تابع `north()` ماشینی که در شمال پل است، قصد ورود به پل را کرده و از پل رد میشود. همچنین با تابع `south()` ماشین های سمت جنوب از پل رد میشوند. سمافور یا میوتکس های مورد نیاز برای حل مسئله را تعریف کرده و دو تابع نامبرده را با استفاده از آنها پیاده سازی کنید.

۴. یک غار تاریخی با نقاشی‌های دیواری باشکوه دارای ورودی بسیار باریکی است که تنها می‌تواند به یک بازدیدکننده اجازه ورود/خروج در هر زمان بدهد.

```
semaphore s1 = // your answer here;
semaphore s2 = // your answer here;

Cave_exploration ( ) {
// your answer here(may contain one or two calls)
Enter_the_cave ( );
// your answer here(may contain one or two calls)
Look_at_the_paintings ( );
// your answer here(may contain one or two calls)
Exit_the_cave ( );
// your answer here(may contain one or two calls)
}
```

برای اطمینان از اینکه هرگز بیش از 15 نفر در غار نباشند، شبه کد روبرو را تکمیل کنید. کد توسط هر فرآیند بازدیدکننده غار اجرا می‌شود. بخش‌های مشخص شده را با مقادیر اولیه سمافور یا فراخوانی‌های سمافور (`wait()`) یا (`signal()`) پر کنید.

نکات تکمیلی

۱. پاسخ‌های خود را در قالب فایل pdf با فرمت زیر ارسال کنید:

HW4_LastName_StudentID که LastName نام خانوادگی شما و StudentID شماره دانشجویی شما است.

۲. انجام این تکلیف به صورت تک نفره است. در صورت مشاهده تقلب، نمرات هم مبدا کپی و هم مقصد آن صفر لحاظ می‌شود.

۳. در صورت وجود ابهام میتوانید با دستیاران آموزشی از طریق تلگرام در ارتباط باشید.

- [hadis ghafouri](#)
- [arash sameni](#)