به نام خدا

تکلیف سوم درس سیستم عامل- همگام‌سازی

دانشگاه صنعتی اصفهان- ترم اول ۱۴۰۰

استاد درس: زینب زالی

۱- برنامه زیر را در نظر بگیرید. این راه حل نرم افزاری برای مساله انحصار متقابل برای دو پردازه پیشنهاد شده است. چهار شرط صحت یک راه حل انحصار متقابل را برای این راه حل بررسی کنید و نشان دهید هر یک صحیح است و یا با بیان یک مثال نقض نشان دهید اشتباه است.

```
boolean blocked [2];
int turn;
void P (int id)
{
 while (true) {
    blocked[id] = true;
    while (turn != id) {
       while (blocked[1-id])
          /* do nothing */;
       turn = id;
    }
    /* critical section */
    blocked[id] = false;
    /* remainder */
 }
}
void main()
{
 blocked[0] = false;
 blocked[1] = false;
 turn = 0;
 parbegin (P(0), P(1));
}
```

۲- میزبان یک مهمانی N>2 نفر مهمان را به خانه دعوت کرده است. میزبان نمی‌خواهد 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 نخی پیاده‌سازی می شوند. میزبان برای ورود همه مهمانان صبر می کند سپس ()openDoor را فراخوانی می کند و یک متغیر شرطی را سیگنال می دهد. 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)
```

۳- مسیله خوانندگان نویسندگان را با **اولویت نویسندگان به خوانندگان** در نظر بگیرید. می‌خواهیم lockهای مخصوصی با نام reader-writer locks داشته باشیم که بتوان در چنین شرایطی از آنها استفاده کرد. سودوکدهای لازم را برای پیاده‌سازی توابع writeLock، readUnlock، readLock و writeUnlock بنویسید. در پیاده‌سازی خود از mutex و conditional variable استفاده کنید (از سمافور استفاده نکنید)

۴- می‌خواهیم گذر ماشین‌ها از روی یک پل یک طرفه را مدیریت کنیم به صورتی که تا وقتی ماشین‌هایی در حال عبور از پل از یک سمت هستند به ماشین‌های دیگر که از سمت دیگر قصد ورود دارند اجازه ورود داده نشود. محدودیتی روی تعداد ماشین‌هایی که روی پل هستند وجود ندارد، اما می‌خواهیم مسأله 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 ماشین‌ها از سمت شمال جلوگیری کرده تا ماشین‌هایی 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 ماشین‌های سمت جنوب از پل رد می‌شوند. سمافور یا میتوکس‌های موردنیاز برای حل مسأله را تعریف کرده و دو تابع نامبرده را با استفاده از آن‌ها پیاده‌سازی کنید.

۵) در مورد روش‌های synchronization در یکی از زبان‌ها یا کتابخانه‌های زیر (به صورتی که مشخص شده است) تحقیق کنید و در هر یک از این زبان‌ها حداقل دو روش را با توضیح نوع کاربرد و نحوه استفاده شرح دهید. همچنین در هر یک توضیح دهید به چه صورت می‌توان مانیتور را پیاده‌سازی کرد.

الف) افرادی که ششمین رقم شماره دانشجویی آنها زوج است: پایتون

ب) افرادی که ششمین رقم شماره دانشجویی آنها فرد است: QT