



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

تکلیف سوم درس سیستم عامل

نیمسال تحصیلی پاییز-۱۴۰۰
مدرس: دکتر محمدرضا حیدرپور

دستیاران آموزشی:
مجید فرهادی - دانیال مهرآیین - محمد نعیمی

نحوه تحویل:

پاسخ های خود به همراه برنامه های نوشته شده را در قالب یک فایل PDF در سامانه بارگذاری کنید. استفاده از \LaTeX اختیاری بوده و ۴۰ نمره اضافی دربرخواهد داشت. می توانید برای آشنایی با دستورات \LaTeX از این قالب آماده شده استفاده کنید.

۱. به سوالات زیر به صورت کوتاه پاسخ دهید. (۵۰ نمره)

(آ) چه تفاوتی هایی بین Thread و Process وجود دارد؟

(ب) چه اطلاعاتی بین Thread های مختلف یک پروسس مشترک نیست؟ چرا؟

(ج) Multithreading چگونه به آسان شدن برنامه نویسی کمک می کند؟

(د) Multithreading در پردازنده هایی که توانایی Concurrency ندارند، چه مزایایی دارد؟

(ه) از دید زمان بندها، Thread های مختلف یک پروسس مستقل هستند؟ چرا؟

(و) کدام یک از قفل های معرفی شده در درس، Starvation ایجاد نمی کنند؟

(ز) در یک سیستم ۴ منبع بین ۳ رشته مختلف به اشتراک گذاشته شده اند. هر یک از این رشته ها حداکثر به ۲ منبع از این ۴ منبع نیاز دارند. با توجه به این مفروضات، امکان وقوع Deadlock در این سیستم وجود دارد؟ چرا؟

۲. با استفاده از Semaphore مسئله Producer-Consumer را در هر کدام از حالات زیر حل نمایید. (۶۰ نمره)

- طول بافر محدود

- طول بافر بی نهایت

۳. برنامه زیر را در نظر بگیرید:

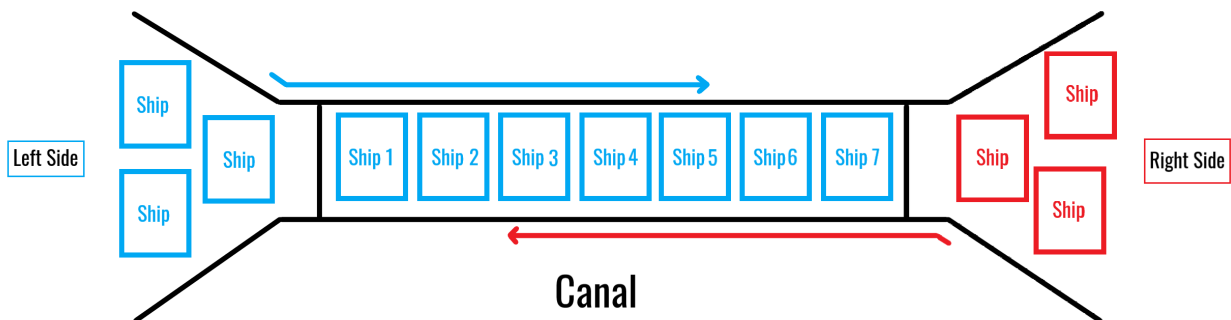
```
do
{
    while (true)
    {
        flag[i] = want_in;
        j = turn;
        while (j != i)
        {
            if (flag[j] != idle)
                j = turn;
            else
                j = (j + 1) % n;
        }
        flag[i] = in_cs;
        j = 0;
        while ( (j < n) && (j == i || flag[j] != in_cs))
            j++;
        if ( (j >= n) && (turn == i || flag[turn] == idle))
            break;
    }
    /* critical section */
    j = (turn + 1) % n;
    while (flag[j] == idle)
        j = (j + 1) % n;
    turn = j;
    flag[i] = idle;
    /* remainder section */
} while (true);
```

که در آن پروسس‌ها متغیرهای زیر را به اشتراک گذاشته‌اند:

```
enum pstate {idle, want_in, in_cs};
pstate flag[n];
int turn;
```

در ابتدا تمامی flagها برابر با idle هستند، turn مقداری بین 0 تا $n - 1$ دارد و i شماره پروسس فعلی است. ثابت کنید روش فوق، مسئله Critical-Section را برای n پروسس حل می‌کند. (۸۰ نمره)

۴. یک کانال کشتی‌رانی به صورت زیر طراحی شده است. عرض این کانال به اندازه عرض 1 کشتی بوده و طول آن می‌تواند 7 کشتی را در خود جای دهد. اپراتور کانال فقط زمانی که تعداد کشتی‌های منتظر در یک سمت کانال به عدد 7 برسد، اجازه ورود آن‌ها را به کانال می‌دهد. همچنین در هر لحظه فقط امکان تردد از یک سمت این کانال وجود دارد.



شکل ۱: کانال کشتی‌رانی

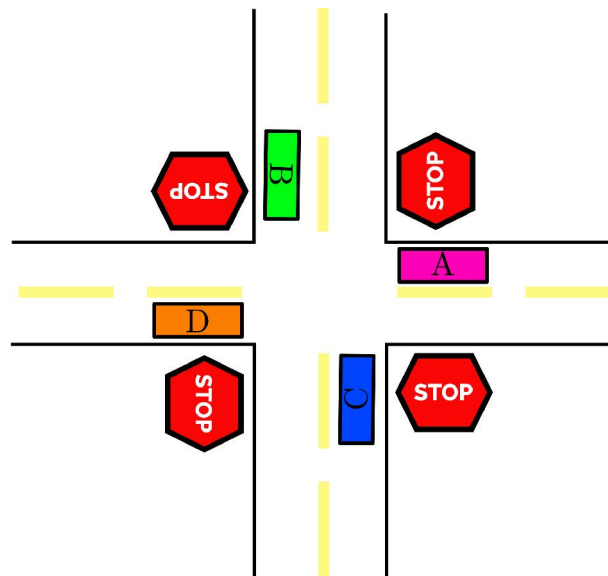
برای هر یک از طرفین کانال توابعی برای ورود به کانال و خروج از آن پیاده‌سازی نمایید:

- Right_Enter
- Right_Exit
- Left_Enter
- Left_Exit

فرض کنید Critical-Section تابعی تحت عنوان Canal_Pass است. (۸۰ نمره)

۵. در این سوال قصد داریم عملکرد قفل‌ها و رخداد Race Condition را بررسی کنیم. ابتدا فایل اجرایی مربوط به شبیه سازی را از اینجا دانلود کرده و سپس با توجه به توضیحات داده شده در این فایل، به سوالات 11 و 12 موجود در انتهای آن پاسخ دهید. همچنین می توانید برای آشنایی با طرز استفاده از این شبیه ساز به راهنمای شبیه ساز مراجعه کنید. تصویر مربوط به نتیجه شبیه سازی را به همراه تحلیل خود از نتیجه آن را در پاسخ نامه خود قرار دهید. (۳۰ نمره)

۶. (اختیاری) Deadlock ترافیکی زیر را در نظر بگیرید:



شکل ۲: چهارراه

- (آ) 4 شرطی که باعث وقوع Deadlock در این مسئله شده‌اند را ذکر کنید. (۱۰ نمره)
- (ب) قوانینی برای جلوگیری Deadlock بدون ایجاد Starvation مطرح نمایید. (۲۰ نمره)
- (ج) برنامه‌ای بنویسید که تعداد خودروها را به عنوان ورودی دریافت، آن‌ها را به طور مساوی بین ورودی‌های چهارراه تقسیم و در نهایت اپراتور این چهارراه را پیاده‌سازی کند. (خودروها Threadهای برنامه هستند). برای ورود هر خودرو عبارت "Car TID is waiting in A/B/C/D side." و برای خروج هر خودرو عبارت "Car TID is exiting from A/B/C/D side." را در خروجی چاپ کند. (TID همان ID مربوط به هر Thread است و A/B/C/D نیز جهت ورود و خروج خودرو است که باید تعیین شود. فرض کنید خودروها مستقیم حرکت می‌کنند). کد خود را به همراه تصاویر خروجی برنامه به ازای ورودی‌های 50 و 100 در پاسخ‌نامه قرار دهید. (۷۰ نمره)

۷. (غیرتحویلی) برخی پیاده‌سازی‌های Semaphore، تابعی به نام `getValue()` دارند که مقدار کنونی آن را برمی‌گرداند. استفاده از این تابع چه مشکلاتی را پدید می‌آورد؟

۸. (غیرتحویلی) Merge Sort موازی‌شده را پیاده‌سازی کنید. آیا با وجود همزمانی، به قفل نیاز است؟ چرا؟

۹. (غیرتحویلی) روش Monte Carlo برای تقریب عدد π را به صورت موازی‌شده پیاده‌سازی کنید.

۱۰. (غیرتحویلی) مشکل برنامه زیر چیست؟ آن را برطرف نمایید.

```
#include <pthread.h>
void *thread(void *vargp)
{
    sleep(1);
    printf("Hello world!\n");
    return NULL;
}
int main()
{
    pthread_t tid;
    pthread_create(&tid, NULL, thread, NULL);
    exit(0);
}
```

موفق باشید.