

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

سیستم‌های عامل (بهار ۱۴۰۱)

تمرین چهارم

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پاسخ:

جمعه ۳۰ اردیبهشت ساعت

۲۳:۵۹

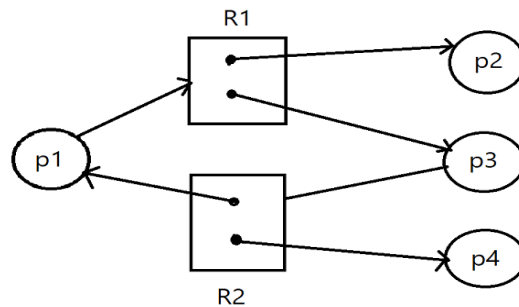
نکته مهم: دقت کنید که تمدید نخواهیم داشت و صرفاً می‌توانید ۱ تا ۵ روز از ۱۵ روز مجاز برای تاخیر ارسال تمامی تمرین‌های تئوری در این ترم را استفاده کنید. اگر بودجه ۱۵ روز شما تمام شود، به ازای هر روز تاخیر ۱۰ درصد از نمره تمرین را از دست خواهید داد.

۱) الف) فرض کنید در مجموع ۹ واحد از یک نوع منبع موجود است، با توجه به اطلاعات داده شده راجع به هر پردازش، نشان دهید که آیا دنباله ای وجود دارد که در حالت safe state باشد؟

*به عنوان مثال دنباله ی (P_4, P_2, P_3, P_1) ، از چپ به راست، در حالت ایمن نیست.

| process | used | Max |
|---------|------|-----|
| P_1 | 2 | 7 |
| P_2 | 1 | 6 |
| P_3 | 2 | 5 |
| P_4 | 1 | 4 |

ب) آیا در گراف زیر حلقه (cycle) وجود دارد؟ deadlock چگونه؟ علت را بیان کنید.



۲) سیستم زیر را در نظر بگیرید ، آیا سیستم در حالت امن است؟ به طور کامل توضیح دهید.

*ستون available بعد از اختصاص دادن منابع به پردازش ها است.

| process | MAX | Allocation | Available |
|---------|---------|------------|-----------|
| | A B C D | | |
| p_0 | 2 1 0 6 | 1 0 0 4 | 1 1 2 3 |
| P_1 | 0 5 7 1 | 0 0 1 1 | |
| P_2 | 6 5 2 3 | 4 5 2 1 | |
| P_3 | 3 5 6 1 | 3 3 6 0 | |
| P_4 | 6 5 6 1 | 2 1 2 0 | |

۳) با توجه به الگوریتم‌های زیر، برای هر کدام مشخص کنید که کدام یک از شروط انحصار متقابل، پیشرفت (progress) و انتظار محدود (bounded waiting) را رعایت می‌کنند و کدام یک را نقض می‌کنند. دلایل خود را بنویسید.

الف) راه حلی برای دو پردازنده

Method used by P₁

```
while(true){  
    while(turn!=1);  
        //critical section  
    turn = 2;  
        //outside of critical section  
}
```

Method used by P₂

```
while(true){  
    while(turn!=2);  
        //critical section  
    turn = 1;  
        //outside of critical section  
}
```

ب) راه حلی برای دو پردازنده

```
do {  
    flag[j] = true;  
    turn = j;  
    while ( flag[i] && turn == j);  
        //critical section  
    flag[j] = false;  
        //remainder section  
} while (true);
```

۴) قطعه کد زیر را در نظر بگیرید. در این تابع قصد داریم عملیات ضرب بین مقدار `op1` و مقداری که `P_op2` به آن اشاره دارد را انجام دهیم. حاصل ضرب باید در حافظه ای که `P_op2` به آن اشاره می‌کند ذخیره شود. این تابع را طوری با دستور `compare_and_swap` کامل کنید که عملیات ضرب به صورت اتمی انجام شود. همچنین برقرار بودن یا نبودن هر کدام از شروط سه گانه را با دلیل شرح دهید.

```
int multiplication(int op1, int *P_op2){
    // TODO
    return *P_op2, *op1;
}
```

پیاده سازی تابع `compare_and_swap` را به صورت زیر در نظر بگیرید:

```
bool compare_and_swap (int *value, int old, int new){
    if(*value != old){
        return false;
    }
    *value = new;
    return true;
}
```

۵) انتظار مشغول (`busy waiting`) چیست؟ سایر انتظارهای موجود در سیستم عامل کدامند؟ آیا به صورت کلی میتوان از انتظار مشغول اجتناب کرد؟ پاسخ خود را توضیح دهید.

موفق باشید

تیم تدریس یاری درس سیستم های عامل