به نام خداوند بخشنده مهربان

سیستم‌های عامل

دانشکده مهندسی کامپیوتر

محمدعلی میرزایی و محدثه میربیگی
پاییز ۱۴۰۲

۳

تمرین سوم تاریخ انتشار

سوالات تمرین سوم

تاریخ تحویل: ۱۱ آذر ۱۴۰۲

پرسش ۱ (۲۰ نمره) برنامه C زیر را در نظر بگیرید

```c
pthread_mutex_t x, y, z;
void *a(void *arg)
{
    pthread_mutex_lock(&y);
    pthread_mutex_lock(&z);
    pthread_mutex_unlock(&y);
    pthread_mutex_unlock(&z);
    return NULL;
}
void *b(void *arg)
{
    pthread_mutex_lock(&z);
    pthread_mutex_lock(&x);
    pthread_mutex_unlock(&z);
    pthread_mutex_unlock(&x);
    return NULL;
}
void *c(void *arg)
{
    pthread_mutex_lock(&x);
    pthread_mutex_lock(&y);
    pthread_mutex_unlock(&x);
    pthread_mutex_unlock(&y);
    return NULL;
}
int main(int argc, char **argv)
{
    pthread_mutex_init(&x, NULL);
    pthread_mutex_init(&y, NULL);
    pthread_mutex_init(&z, NULL);
    pthread_t a_thread, b_thread, c_thread;
    pthread_create(&a_thread, NULL, a, NULL);
    pthread_create(&b_thread, NULL, b, NULL);
    pthread_create(&c_thread, NULL, c, NULL);
    pthread_join(&a_thread, NULL);
    pthread_join(&b_thread, NULL);
    pthread_join(&c_thread, NULL);
    return 0;
}
```

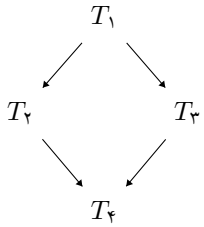
(آ) یک ترتیب اجرایی (از اجرای تردها و اخذ و رها کردن لاک‌ها) ارائه دهید که برنامه دچار Deadlock شود.

(ب) آیا با تغییر ترتیب mutex_unlockها می‌توان از رخ دادن Deadlock جلوگیری کرد؟ در صورت امکان این کار را انجام دهید یا دلیل عدم امکان را بنویسید.

(ج) در صورتی که از یک زمان‌بند FCFS استفاده کنیم، آیا امکان رخ دادن Deadlock وجود دارد؟ توضیح دهید.

(د) در صورت امکان با بازنویسی یکی از سه تابع a،b،c Deadlock را برطرف کنید. دلیل برطرف شدن Deadlock یا عدم امکان را توضیح دهید.

پرسش ۲ (۱۰ نمره) گراف زیر را در نظر بگیرید که ارتباط میان چهار ترد $(T_۱, T_۲, T_۳, T_۴)$ را نشان می‌دهد. یک فلش از یک ترد مانند T_x به ترد دیگر مانند T_y به این معناست که ترد T_x باید Computation خود را قبل از این که T_y شروع کند، به اتمام برساند. بدون فرضی راجع‌به ترتیب زمانبندی ریسه‌ها، از حداقل تعداد سمافور استفاده کنید تا مطمئن شوید تردها ترتیب نشان‌داده شده در گراف را عملی می‌کنند. سمافورها را مقداردهی اولیه کرده و مکان عملیات‌های مربوط به سمافورها را در توابع زیر مشخص کنید.



// Semaphores definitions and their initial values			
void $T_۱$(void) {	void $T_۲$(void) {	void $T_۳$(void) {	void $T_۴$(void) {
// $T_۱$ Computation	// $T_۲$ Computation	// $T_۳$ Computation	// $T_۴$ Computation
}	}	}	}

پرسش ۳ (۵ نمره) الگوریتم Banker برای نگه داشتن یک سیستم در حالت **safe** به‌کار برده می‌شود. توضیح دهید که حالت safe به چه معناست و مختصرا بگویید الگوریتم Banker به چه شکل سیستم را در حالت safe نگه می‌دارد؟

پرسش ۴ (۲۰ نمره) فرض کنید که شما در حال نوشتن کد یک سیستم multiprocessor هستید که از تردها استفاده می‌کند. می‌خواهیم کمی با مفاهیم Semaphoreها و Monitorها بیش‌تر آشنا شویم. فرض کنید که interface یک سمافور به شکل زیر باشد:

```
public class Semaphore {
    public Semaphore(int initialValue) {
        /* Create and return a semaphore with initial value: initialValue */
        ...
    }
    public P() {
        /* Call P() on the semaphore */
        ...
    }
    public V() {
        /* Call V() on the semaphore */
        ...
    }
}
```

همان‌طور که می‌دانید، یک Monitor از یک یا بیش‌تر Condition Variable تشکیل شده‌است. در زیر interface های این دو مورد را آورده‌ایم:

```
public class CondVar {
    public CondVar(Lock lock) {
        /* Creates a condition variable associated with Lock lock. */
        ...
    }
    public void Wait() {
        /* Block on condition variable */
        ...
    }
    public void Signal() {
        /* Wake one thread (if it exists) */
        ...
    }
    public void Broadcast() {
        /* Wake up all threads waiting on cv*/
        ...
    }
}
```

```
public class Lock {
    public Lock() {
        /* Create new Lock */
        ...
    }
    public void Acquire() {
        /* Acquire Lock */
        ...
    }
    public void Release() {
        /* Release Lock */
        ...
    }
}
```

Monitorها و Semaphoreها هر کدام می‌توانند توسط دیگری پیاده‌سازی شوند. در این سوال معادل بودن این دو مورد را بررسی می‌کنیم.

(آ) تفاوت میان زمانبندی **Mesa** و **Hoare** در Monitorها را بیان کنید.

(ب) با استفاده از توابع کلاس‌های Monitor (یعنی کلاس‌های Lock و CondVar) کلاس Semaphore را پیاده‌سازی کنید.(هر سه تابع Semaphore()، P()، V() را در حداکثر ۵ خط تکمیل کنید.) فرض کنید که Monitorها بر پایه‌ی Mesa هستند.

(ج) با استفاده از کلاس Semaphore، کلاس Lock را پیاده‌سازی کنید. (هر سه تابع Lock()، Acquire()، Release() را در حداکثر ۵ خط تکمیل کنید.)

(د) تفاوت میان Semaphore.V() و CondVar.Signal() را زمانی که هیچ تردی برای سمافور یا condition variable مربوطه منتظر نیست، شرح دهید.

پرسش ۵ (۲۰ نمره) فرض کنید پردازه‌ای با ۳ thread که به صورت همروند اجرا می‌شوند در اختیار داریم. برای همگام‌سازی این threadها از ۳ سمافور استفاده شده است. عملیات اجرایی توسط هر thread در شبه‌کد زیر قابل مشاهده است. با توجه به این شبه کد به سوالات زیر پاسخ دهید.

```
// Thread 1
L1:
    wait(semaphore_1);
    printf("3");
    signal(semaphore_3);
    goto L1;

// Thread 2
L2:
    wait(semaphore_2);
    printf("2");
    signal(semaphore_1);
    goto L2;

// Thread 3
L3:
    wait(semaphore_3);
    printf("1");
    signal(semaphore_2);
    goto L3;
```

(آ) آیا می‌توان سمافورها را طوری مقداردهی اولیه کرد که threadها در طی اجرای همروند خود حتما رشته‌ای با الگوی ۱۲۳۱۲۳۱۲۳۱۲۳۱۲ در ابتدای خود را تولید کنند؟ توضیح دهید.

(ب) در صورتی که مقادیر اولیه سمافورها به صورت $\text{semaphore\_1} = 1, \text{semaphore\_2} = 2, \text{semaphore\_3} = 6$ باشد، آیا امکان دارد که این threadها به نحوی اجرا شوند که رشته تولیدی با الگوی ۱۱۳۳۲۳۳ آغاز شود؟ توضیح دهید.

پرسش ۶ (۱۵ نمره) فرض کنید که تصویری از یک لحظه خاص از منابع یک سیستم و وضعیت اختصاص آنها به پردازه‌های موجود، در جدول ۱ به شما داده شده است. به سوالات زیر پاسخ دهید.

جدول ۱: تصویر لحظه‌ای اختصاص منابع سیستم

Total Resources			Maximum			Allocation			Process
C	B	A	C	B	A	C	B	A	
۱۲	۹	۱۲							
			۴	۹	۴	۳	۱	۲	P1
			۳	۳	۵	۳	۲	۱	P2
			۳	۴	۶	۳	۴	۵	P3
			۲	۸	۴	۲	۱	۲	P4

(آ) آیا در این لحظه سیستم در حالت امن قرار دارد؟ توضیح دهید.

(ب) اگر در اولین لحظه بعد از شروع از وضعیت جدول داده‌شده، پردازه P1 درخواست ۲ واحد از منبع A را بدهد، آیا سیستم می‌تواند با پاسخ به این درخواست در همان لحظه و اعطای منبع، همچنان بدون بن‌بست به اجرای پردازه‌ها ادامه بدهد؟ توضیح دهید.

(ج) در صورتی که از همان وضعیت اولیه جدول شروع کنیم، پردازه P1 در همان ابتدا اجازه درخواست حداکثر چند واحد از هر کدام از سه منبع موجود در سیستم را به صورت (A, B, C) دارد، به طوری که سیستم پس از پاسخ به درخواست همچنان در یک حالت امن باشد؟ توضیح دهید.

پرسش ۷ (۱۰ نمره)

درستی یا نادرستی عبارت‌های زیر را توضیح دهید.

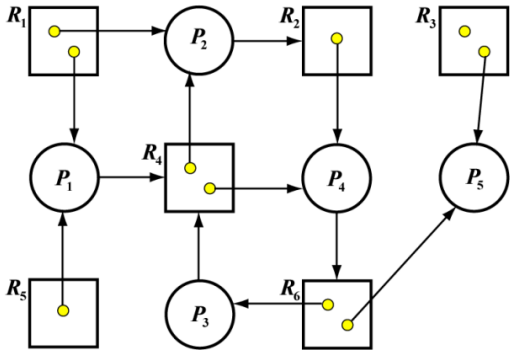
(آ) اگر در یک سیستم به هر پردازه یک پردازنده اختصاص داده شود، در این سیستم مشکل Deadlock ایجاد نخواهد شد.

(ب) اگر در یک سیستم به Deadlock خورده باشیم، از بین بردن هر پردازه دلخواهی در این سیستم باعث بیرون آمدن از Deadlock خواهد شد.

(ج) اگر در یک سیستم هر پردازه قبل از اینکه آغاز به اجرا کند، همه منابعی را که برای اجرا لازم دارد قفل کند، در این سیستم هیچ Deadlock ایجاد نخواهد شد.

(د) با استفاده از اولویت‌بندی پردازه‌ها در یک سیستم، می‌توان از رفتن سیستم به Deadlock جلوگیری کرد.

پرسش ۸ (۰ نمره) این سوال تحویل دادنی نیست و برای آشنایی شما با این نوع از سوالات است. گراف تخصیص منابع (Resource Allocation Graph) یک سیستم به صورت زیر است:



- ابتدا این گراف را به صورت ماتریسی نشان دهید. (یعنی ماتریس تخصیص، موجود و درخواست این سیستم را رسم کنید.)
- آیا این سیستم در بن‌بست (Deadlock) قرار دارد؟ در صورتی که این سیستم در بن‌بست قرار دارد نشان دهید کدام پردازه‌ها باعث این موضوع شده‌اند و اگر هم بن‌بست وجود ندارد، یک حالت امن (Safe State) از این سیستم را بنویسید.