تکلیف اول سیستم‌های عامل

دکتر زینب زالی

دانشکده مهندسی برق و کامپیوتر

تاریخ تحویل: آبان ۱۴۰۲

---

۱. توضیح دهید چرا استفاده از روش غیرفعال‌سازی وقفه برای پیاده‌سازی اصول همزمانی در سیستم‌های چندهسته‌ای مناسب نمی‌باشد؟

۲.

الف) در هرکدام از شرایط زیر، استفاده از spin lock بهتر است یا mutex lockای که پروسس در هنگام منتظر ماندن به خواب می‌رود؟

(در حالت spin lock یک پروسس به‌جای اینکه به حالت خواب برود، با چرخش در یک حلقه منتظر می‌ماند تا قفل آزاد شود)

- قفل برای مدت کوتاهی نگه‌داری می‌شود.
- قفل برای مدت طولانی نگه‌داری می‌شود.
- یک thread ممکن‌است هنگامی که قفل را در اختیار دارد به خواب برود.

ب) اگر context switch به مدت زمان T نیاز داشته‌باشد، یک upper bound برحسب T برای استفاده از spin lock مشخص کنید.

۳. توضیح دهید چرا استفاده از spinlock در سیستم‌های سینگل پروسسور مناسب نمیباشد.

۴. اولین راه حل نرم‌افزاری شناخته‌شده برای مسئله Critical-Section برای دو پروسه توسط Dekker ارائه داده‌شده‌است. کد الگوریتم برای دو پروسه p0 و p1 در زیر نشان داده شده است. ثابت کنید که الگوریتم هر ۳ شرط مسئله Critical-Section را ارضا می‌کند؟

```
p0:
   wants_to_enter[0] ← true
   while wants_to_enter[1] {
      if turn ≠ 0 {
         wants_to_enter[0] ← false
         while turn ≠ 0 {
           // busy wait
         }
         wants_to_enter[0] ← true
      }
   }

   // critical section
   ...
   turn ← 1
   wants_to_enter[0] ← false
   // remainder section
```

```
p1:
   wants_to_enter[1] ← true
   while wants_to_enter[0] {
      if turn ≠ 1 {
         wants_to_enter[1] ← false
         while turn ≠ 1 {
           // busy wait
         }
         wants_to_enter[1] ← true
      }
   }

   // critical section
   ...
   turn ← 0
   wants_to_enter[1] ← false
   // remainder section
```

۵. اولین راه حل نرم‌افزاری شناخته‌شده برای مسئله Critical-Section برای n پروسه توسط McGuire & Eisenberg ارائه داده‌شده‌است. درباره این الگوریتم تحقیق کرده و ثابت کنید این الگوریتم سه شرط مسئله Critical-Section را ارضا می‌کند.

۶. مشکلی که Reordering در الگوریتم پترسون ایجاد می‌کند را توضیح داده و از دستور memory_barrier() استفاده نمایید تا مشکل الگوریتم را حل کنید.

۷. میخواهیم Mutex Lock را با استفاده از دستور test_and_set() پیاده‌سازی کنیم. از استراکت زیر استفاده کرده و دو تابع void acquire(lock *mutex) و void release(lock *mutex) را برای آن به صورت سودوکد پیاده‌سازی نمایید.

```
typedef struct{
    int available;
} lock;
```

۸. کد زیر را در نظر بگیرید، دو ترد میخواهند همزمان تابع calc را اجرا کنند. ترد اول آرایه A و ترد دوم آرایه B را به عنوان ورودی به تابع calc میفرستد. مقدار نهایی critical چه عددی است؟ تنها با تغییر جای mutex.acquire و mutex.release کد را بهینه‌تر کنید و توضیح دهید چرا کد شما سریعتر اجرا می‌شود؟

```
int critical = 0;
int *A = {1, 2, 3, 4};
int *B = {5, 6, 7, 8};
void *calc(int *array)
{
    mutex.acquire();
    int temp = 0;
    for (int i = 0; i < 4; i++)
    {
        temp += array[i];
    }
    critical += temp;
    mutex.release();
}
```

۹. سرورها می‌توانند گونه‌ای طراحی شوند که تعداد اتصالات باز را محدود کنند. به عنوان مثال، یک سرور ممکن است بخواهد در هر لحظه حداکثر N اتصال سوکت داشته باشد. به محض اینکه N اتصال برقرار شد، سرور تا زمانی که یک اتصال موجود قطع نشود، اتصال دیگری را نخواهد پذیرفت. توضیح دهید چگونه می‌توان از سمافورها استفاده کرد تا تعداد اتصالات همزمان را محدود کرد.

نکات تکمیلی

۱.فرمت نام گذاری تکلیف ارسالی باید به صورت زیر باشد:
HW3 _LastName_StudentID که LastName نام خانوادگی شما و StudentID شماره دانشجویی شما است.
۲. انجام این تکلیف به صورت تك نفره است. در صورت مشاهده تقلب، نمرات هم مبدا کپی و هم مقصد آن صفر لحاظ می شود.
۳. در صورت وجود ابهام میتوانید با دستیاران آموزشی از طریق تلگرام در ارتباط باشید.

- hadis ghafouri
- arash sameni