

تمرینات فصل 3 سیستم عامل

لطفا پاسخ تمرینات را علاوه بر بارگذاری در سامانه LMS ، در زمان مقرر به آدرس ostbzu14032@gmail.com نیز ارسال کنید.

1- کد زیر برای چاپ اعداد ۰ تا ۹ با استفاده از دو Thread نوشته شده. اما در اجرا معمولاً تعدادی عدد حذف می‌شوند یا ترتیب آن‌ها به هم می‌ریزد. اشکال را مشخص کرده و راه حل بدهید.

```
DECLARE i = 0

FUNCTION print_numbers()
  WHILE i < 10 DO
    PRINT i
    i = i + 1
  END WHILE
END FUNCTION

MAIN:
  CREATE THREAD t1 WITH FUNCTION print_numbers
  CREATE THREAD t2 WITH FUNCTION print_numbers
  WAIT FOR t1 TO FINISH
  WAIT FOR t2 TO FINISH
END MAIN
```

2- فرض کنید سیستمی دارای ۸ پردازنده با اولویت‌های مختلف است که برخی از آن‌ها از منابع مشترک استفاده می‌کنند. چگونه استفاده از Thread Synchronization می‌تواند منجر به **Starvation** شود؟ چه راهکارهایی برای جلوگیری از آن پیشنهاد می‌کنید؟

3- دو مدل رایج پیاده‌سازی thread عبارتند از: مدل user-level threads و kernel-level threads. با در نظر گرفتن سناریوهای پیچیده مانند blocking I/O و سیستم‌های تعبیه‌شده (embedded) ، تحلیل کنید که هر مدل در چه شرایطی مناسب‌تر است. همچنین به بحث هزینه‌ی context switch در هر مدل اشاره کنید.

4- در یک سیستم با چند Thread فعال، شرایطی پیش آمده که CPU استفاده نمی‌شود (idle) در حالی که هیچ Thread ی در حالت پایان‌یافته یا مسدود (blocked) نیست. همه‌ی Thread ها در حالت آماده (runnable) هستند، ولی هیچ‌کدام اجرا نمی‌شوند. با توجه به مفاهیم Thread Synchronization و Scheduler، چه شرایطی می‌تواند باعث چنین وضعیتی شود؟ آیا این می‌تواند ناشی از Starvation یا Deadlock باشد؟ تحلیل کنید و مثالی فرضی بزنید که این وضعیت را ایجاد کند.

5- در یک برنامه‌ی چندنخی (multi-threaded)، ۲۰٪ از کد قابل موازی‌سازی نیست. فرض کنید می‌خواهیم با استفاده از ۴ هسته این برنامه را اجرا کنیم. با استفاده از قانون Amdahl، سرعت‌بخشی (speedup) حداکثری برنامه را محاسبه کنید. سپس تحلیل کنید چرا حتی با افزایش تعداد هسته‌ها به بی‌نهایت، باز هم نمی‌توان اجرای برنامه را آن‌طور که انتظار داریم سریع کرد. این موضوع چه تأثیری روی طراحی الگوریتم‌های هم‌زمان دارد؟

6- در یک سیستم تک‌هسته‌ای، چند Thread با اولویت‌های مختلف اجرا می‌شوند. برخی Thread ها در حال انجام محاسبات CPU-bound هستند و برخی I/O-bound هستند. توضیح دهید چگونه در این سیستم، بدون وجود Parallelism واقعی، همچنان می‌توان Concurrency را تجربه کرد. در پاسخ خود به نقش Scheduler و نحوه مدیریت زمان‌بندی بین Thread های I/O-bound و CPU-bound اشاره کنید. همچنین توضیح دهید چرا یک Thread ممکن است «هم‌زمان» به نظر برسد ولی واقعاً به‌طور موازی اجرا نمی‌شود.

7- فرض کنید برنامه‌ای دارید که یک ماتریس بزرگ را پردازش می‌کند و این پردازش به بخش‌های مستقل تقسیم‌پذیر است. سیستم دارای ۸ هسته واقعی (real cores) است. با افزایش تعداد Thread به ۸، سرعت اجرای برنامه بهتر می‌شود، اما وقتی به ۱۶ می‌رسد، به‌جای بهبود، زمان اجرای کل افزایش می‌یابد. با توجه به مفهوم Parallelism، توضیح دهید چرا افزایش تعداد Thread ها پس از حد مشخص باعث افت عملکرد می‌شود. چه عواملی در این پدیده نقش دارند؟ در پاسخ خود به مواردی مثل context switch، overhead، cache contention و thread scheduling اشاره کنید.

(سوالات امتیازی)

سه Thread وجود دارد به نامهای $T1$ ، $T2$ ، و $T3$. هدف این است که آنها به

ترتیب خاصی اجرا شوند: $T1 \rightarrow T2 \rightarrow T3$ و این ترتیب باید در یک حلقه ۵ بار تکرار شود.

از Mutex و Condition Variable استفاده کنید. بر اساس سودوکد زیر، جای بخش‌های خالی را طوری کامل کنید که ترتیب اجرای دقیق حفظ شود و هیچ Race Condition یا Deadlock رخ ندهد.

```

// Shared variables
int turn = 1
mutex lock
condition cond

function T1() {
    repeat 5 times {
        lock.acquire()
        while (turn ≠ 1)
            cond.wait(lock)

        print("T1 running")

        // تکمیل شود
        _____

        cond.signal()
        lock.release()
    }
}

function T2() {
    repeat 5 times {
        lock.acquire()
        while (turn ≠ 2)
            cond.wait(lock)

        print("T2 running")

        // تکمیل شود
        _____

        cond.signal()
        lock.release()
    }
}

function T3() {
    repeat 5 times {
        lock.acquire()
        while (turn ≠ 3)
            cond.wait(lock)

        print("T3 running")

        // تکمیل شود
        _____

        cond.signal()
        lock.release()
    }
}
```

کد زیر قصد دارد دو Thread را ایجاد کند که به نوبت عددی را از آرایه چاپ کنند. اما گاهی ترتیب چاپ اشتباه می‌شود. اشکال را پیدا و اصلاح کنید.

```
DECLARE arr = [1, 2, 3, 4]
DECLARE index = 0

FUNCTION print()
    WHILE index < 4 DO
        PRINT arr[index]
        index = index + 1
    END WHILE
END FUNCTION
```