آزمون میانی درس سیستم عامل

مجموع نمره‌ها: ۱۵ + ۱۰۰ زمان آزمون: ۸۰ دقیقه

---

۱ (۵) برنامه‌ی شما و یکی از دوستانتان در یک سیستم عامل چند کاربره همزمان اجرا می‌شوند. او شکایت می‌کند که برنامه‌ی شما قسمتی از حافظه‌ی برنامه‌ی او را تغییر داده است و موجب شده است برنامه‌ی او دچار خطا شود. چه دلیلی بر خلاف این ادعا وجود دارد؟

۲ (۱۰) یکی از دوستانتان ادعا می‌کند که واقعا احتیاجی به سیستم عامل نیست و راه‌اندازهای سخت‌افزاری را می‌توان در یک کتابخانه قرار داد تا برنامه‌نویسان کاربردی در برنامه‌هایشان از آن استفاده کنند. الف) مزیت این کار چیست؟ ب) با دلیل پاسخ دهید: آیا این کتابخانه نیاز به سیستم عامل را از بین می‌برد؟

۳ (۱۰) یکی از برنامه‌هایتان گاهی کمی کندتر از حالت عادی اجرا می‌شود. چگونه سخت‌افزار (کارت شبکه‌ای با نرخ بالای ترافیک) یا سایر برنامه‌های در حال اجرا می‌توانند در این کاهش سرعت مؤثر باشند؟ برای هر یک از این دو عامل، با دقت بیان کنید چگونه و به چه علتی پردازنده از پردازه‌ی شما گرفته می‌شود.

۴ (۵) در یک سیستم عامل، در چه هنگام بهتر است درجه‌ی چند برنامگی (تعداد برنامه‌های در حال اجرا) کاهش یابد؟ چه قسمتی از سیستم عامل این تصمیم را می‌گیرد؟

۵ (۵) سیستم عامل، یکی از برنامه‌های شما را به دلیل خطا (دسترسی به قسمت غیر مجازی از حافظه) خاتمه داده است (به عبارت دیگر برنامه‌ی شما Crash کرده است). آیا راهی برای پی بردن به دلیل خطا و حل آن وجود دارد؟ توضیح دهید.

۶ (۲۰) در برنامه‌ای توسط حلقه‌ی زیر تابع compute صد بار صدا زده می‌شود. الف) این قسمت از کد را تغییر دهید که پس از ساختن یک پردازه (با تابع (fork()، نیمی از فراخوانی‌های تابع compute() در پردازه‌ی اول و بقیه در پردازه‌ی جدید انجام شود (بین فراخوانی‌ها وابستگی وجود ندارد). ب) این تغییر چگونه به استفاده‌ی بهتر از یک پردازنده‌ی چند هسته‌ای کمک می‌کند؟ ج) سه راه برای انتقال اطلاعات از پردازه‌ی جدید به پردازه‌ی اول پیشنهاد کنید (فقط نام ببرید).

```
for (i = 0; i < 100; i++)
      compute(i);
```

۷ (۱۵) درخواست‌های فرستاده شده به یک سرور در یک صف قرار می‌گیرند (درخواست‌های جدید به آخر این صف اضافه می‌شوند). این سرور همواره درخواست آخر صف را بر می‌دارد و به آن پاسخ می‌دهد. الف) امکان بروز چه مشکلی وجود دارد (فرض کنید بین درخواست‌ها وابستگی وجود ندارد)؟ ب) اگر سرور پس از هر بار برداشتن درخواست از آخر صف، یک درخواست از اول صف هم بردارد، آیا این مشکل برطرف می‌گردد؟ توضیح دهید.

۸ (۱۵) در یکی از فراخوانی‌های تابع زیر، این تابع مقدار یک را بر می‌گرداند و برای سایر فراخوانی‌ها مقدار صفر را بر می‌گرداند (فرض کنید مقدار اولیه‌ی متغیر isfree صفر است). اگر این تابع توسط چند ریسمان صدا زده شود، آیا امکان رخداد وضعیت رقابتی وجود دارد؟ اگر بله، نشان دهید چگونه این اتفاق می‌افتد؟

```
int allocate(void)
{
      if (isfree == 0) {
            isfree = 1;
            return 1;
      }
      return 0;
}
```

۹ (۳۰) دو تابع increase() و decrease() توسط چند ریسمان صدا زده می‌شوند و موجودی یک حساب را افزایش و کاهش می‌دهند. تابع increase(N) مقدار موجودی حساب را به مقدار N واحد افزایش می‌دهد. همچنین، تابع decrease(N)، موجودی حساب را N واحد کاهش می‌دهد. در صورتی که موجودی حساب به اندازه‌ی کافی نباشد (کمتر از N باشد)، تابع decrease(N) باید منتظر شود تا موجودی حساب حداقل N شود و سپس عمل کاهش موجودی را انجام دهد. این دو تابع را به کمک سمافور یا مانیتور پیاده‌سازی کنید.

واژه‌نامه

برنامه‌نویسان کاربردی..... Application programmers
پردازنده ..... Processor
پردازه..... Process
چند کاربره ..... Multi-user
درجه‌ی چند برنامگی ..... Multi-programming degree
راه‌انداز..... Driver
سرور ..... Server
سمافور..... Semaphore
صف..... Queue
مانیتور..... Monitor
وضعیت رقابتی..... Race condition
همروند..... Concurrent