تمرین شماره ۳ **برنامه‌نویسی موازی**

نیم‌سال اول **۹۹-۹۸**

زمان آپلود: شنبه ۳۰ آذر

تمرین ۱: زمان‌بندی ریسه‌ها

برنامه exer01.cpp را در نظر بگیرید. این برنامه ۳۲ ریسه ایجاد می‌کند. برای مشخص شدن تاثیر زمان‌بندی هر ریسه یک کار مشخص را انجام می‌دهد. برای اطمینان از این‌که تمام ریسه‌ها ایجاد شده‌اند، تابع sleep (3) اجرا شده است. برنامه را چندبار اجرا کنید. آیا ترتیب زمان‌بندی ریسه‌ها توسط سیستم عامل یکسان است؟ دلیل آن را توضیح دهید.

تمرین ۲: ارسال پارامتر به یک ریسه

برنامه‌های exer02_1.cpp و exer02_2.cpp را کامپایل و اجرا کنید. در این دو برنامه روش صحیح ارسال یک پارامتر به ریسه و نیز ارسال چند پارامتر به ریسه از طریق struct نشان داده شده است. حال برنامه exer02_3.cpp را کامپایل و اجرا کنید. کجای این برنامه اشکال دارد؟ چگونه می‌توانید آن را تصحیح کنید؟

تمرین ۳: خروج از یک ریسه

برنامه exer03.cpp را کامپایل و اجرا کنید. خروجی برنامه چیست؟ چرا؟ چگونه می‌توانید آن را تصحیح کنید؟

تمرین ۴: دسترسی دو به دو ناسازگار ریسه‌ها

برنامه exer04_1.cpp ضرب داخلی دو بردار را به صورت سریال نشان می‌دهد. این برنامه را کامپایل و اجرا کنید. خروجی برنامه چیست؟ برنامه exer04_2.cpp ضرب داخلی دو بردار را به صورت موازی نشان می‌دهد. برای دسترسی دو به دو ناسازگار ریسه‌ها به متغیر سراسری sum از mutex استفاده شده است. این برنامه را چند بار اجرا کنید. آیا ترتیب به هنگام‌سازی این متغیر در اجراهای مختلف یکسان است؟ برنامه exer04_3.cpp یک نسخه دیگر از ضرب داخلی دو بردار را به صورت موازی نشان می‌دهد. این برنامه را چند بار اجرا کنید و خروجی برنامه را مشاهده کنید. آیا خروجی برنامه صحیح است؟ چرا؟ چگونه می‌توانید آن را تصحیح کنید؟

تمرین ۵: استفاده از متغیرهای شرطی

برنامه exer05_1.cpp نحوه استفاده صحیح از متغیرهای شرطی را نشان می‌دهد. کد این برنامه را مطالعه کنید و عمل‌کرد این برنامه را توضیح دهید. این برنامه را کامپایل و اجرا کنید. خروجی ریسه‌های این برنامه چیست؟ برنامه exer05_2.cpp استفاده اشتباه از متغیرهای شرطی را نشان می‌دهد. چرا این برنامه به صورت صحیح کار نمی‌کند؟ چگونه می‌توانید آن را تصحیح کنید؟