



دانشگاه صنعتی مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر

در صورت داشتن سوال در مورد این

تمرین، سوال خود را با موضوع تمرین

با ایمیل زیر در میان بگذارید:

OsFall1400@gmail.com

تمرین ششم درس سیستم عامل

مهلت تحویل ساعت ۵۹:۲۳ روز جمعه ۲۱ آبان ۱۴۰۰

تمرینات را انفرادی حل کرده و در سایت مودل (courses.aut.ac.ir) با

قالب زیر بارگذاری نمایید:

StudentID_Name_Last Name

۱- به سوالات زیر پاسخ کوتاه دهید:

الف) در ساختار کلاس *MulticastSocket* جاوا از چه نوع سوکت دیگری استفاده شده است؟

ب) در صورتی که یک فرایند از پورت 1600 میزبان (*host*) اتصال با یک وب سرور برقرار کرده باشد و همزمان فرایند دیگری روی

همان میزبان قصد برقراری ارتباط با همان وب سرور را داشته باشد، چه محدودیتی در اختصاص پورت به فرایند جدید داریم؟

ج) چرا استفاده از سوکت، روشی سطح پایین برای ارتباط بین فرایندهای توزیع شده در نظر گرفته میشود؟ چه روش سطح بالاتری را

برای این منظور می توان نام برد؟

۲- در هر یک از شرایط زیر توضیح دهید استفاده از کدام نوع از *pipe* ها مناسب است؟ (*ordinary pipe, named pipe*)

الف) پس از پایان ارتباط دیگر نمی خواهیم به *pipe* دسترسی داشته باشیم.

ب) می خواهیم از *pipe* بین شبکه ای از فرایندها استفاده کنیم.

ج) می خواهیم امکان ارتباط دو طرفه (*bidirectional*) فراهم باشد.

د) می خواهیم تنها والد و فرزند به آن دسترسی داشته باشند.

۳- می دانیم ۱۰٪ از یک الگوریتم سریال و ۹۰٪ آن موازی است. برای دستیابی به ۸۰٪ از حداکثر میزان تسریع تئوری (*theoretical*

speedup) باید حداقل از چه تعداد ریسمان برای اجرای این الگوریتم استفاده کرد؟

۴- مدل های مختلفی برای ارتباط میان ریسمان های هسته و ریسمان های کاربر وجود دارد؛

الف) در چه مدلی مدیریت ریسمان ها بهینه تر (*efficient*) می باشد؟ در کدام مدل محدودیتی برای تعداد ریسمان های هسته وجود

ندارد؟ آیا این مدل ها در سیستم عامل های امروزی کاربرد دارند؟ چرا؟

ب) در مدل یک به یک (*one-to-one*)، توسعه دهنده باید در چه موردی احتیاط کند؟ چرا؟

ج) مزیت های مدل چند به چند (*many-to-many*) چیست؟ چرا در سیستم های عامل امروزی استفاده از این مدل رواج ندارد؟