

تمرین سری چهارم درس سیستم های عامل

موعد تحویل: ۹۴/۸/۱۵

این تمرین را به صورت گروه های دو نفره انجام دهید.
برنامه ها و توضیحات این تمرین را فقط یکی از افراد گروه در سایت درس آپلود کند.
در صورت مشاهده ی هر گونه شباهت بین کدها نمره ی هر دو گروه ۰ در نظر گرفته می شود.
پاسخ سوالات توضیحی را فقط به زبان فارسی بنویسید.
توضیحات را در قالب یک فایل PDF به همراه کد در یک فایل Zip قرار داده و نام آنرا با قالب زیر ثبت کنید و در سایت درس آپلود کنید.

9231000_NAME1_9131000_NAME2_HW2.zip

۱- یک مثال از دو برنامه A و B بیاورید که در آن شرایط مسابقه وجود داشته باشد، دو حالت از اجرای این دو برنامه را ایجاد کنید که نشان دهنده ی ایجاد شرایط مسابقه است (می توانید از دستور sleep() برای این منظور بهره ببرید. توجه نمایید که مثالهای ارائه شده برای هر گروه باید یکتا باشد).

۲- در این برنامه مساله ی Reader-Writer پیاده سازی خواهد شد. در این مساله دو فرآیند Reader مقدار نوشته شده در یک میانگیر (بافر) مشترک را می خوانند و یک Writer مقدار موجود در میانگیر را تغییر میدهد. میانگیر، متغیری به نام VAL دارد که در ابتدا مقدار "۰" دارد. در هر بار دسترسی، Writer مقدار موجود در میانگیر را خوانده و به همراه PID خود آن را چاپ می کند. علاوه بر این، مقدار موجود در میانگیر را یکی زیاد می کند. هر Reader نیز همواره مقدار میانگیر را خوانده و به همراه PID خود چاپ می کند. Reader و Writer باید به صورت فرآیندهای جداگانه پیاده سازی شوند. برای همگام سازی فرآیندهای Reader و Writer از روش سمافور استفاده کنید. زمانی که مقدار VAL به ۱۰ برسد، برنامه باید خاتمه یابد. توجه کنید که هر دو Reader می توانند به صورت همزمان به میانگیر دسترسی داشته باشند. اولین فرآیند Reader که اقدام به خواندن از میانگیر کند، باید آن را برای خواندن Lock نماید. در زمانی که میانگیر برای خواندن Lock شده باشد، Reader دیگر هم می تواند محتوای آن را بخواند. آخرین فرآیند Reader پس از خواندن میانگیر، Lock را آزاد می کند. فرآیند Writer تنها زمانی قادر به نوشتن در میانگیر است که هیچ کدام از فرآیندهای Reader در حال خواندن آن نباشند. علاوه بر این زمانی که Writer در حال نوشتن روی میانگیر است هیچ کدام از فرآیندهای Reader نباید میانگیر را بخوانند.
با استفاده از تابع sleep() تاخیر مناسب برای مشاهده ی پیام ها را ایجاد کنید.

نمونه ای از اجرای برنامه به صورت زیر باشد:

```
The first reader acquires the lock.
The reader (2716) reads the value 0
The reader (2717) reads the value 0
The last reader releases the lock.
The writer acquires the lock.
The writer (2715) writes the value 1
The writer releases the lock.
The writer acquires the lock.
The writer (2715) writes the value 2
The writer releases the lock.
The first reader acquires the lock.
The reader (2716) reads the value 2
The last reader releases the lock.
The first reader acquires the lock.
The reader (2717) reads the value 2
The last reader releases the lock.
The first reader acquires the lock.
The reader (2716) reads the value 2
The last reader releases the lock.
The first reader acquires the lock.
The reader (2717) reads the value 2
The last reader releases the lock.
The writer acquires the lock.
The writer (2715) writes the value 3
The writer releases the lock.
The writer acquires the lock.
The writer (2715) writes the value 4
The writer releases the lock.
The first reader acquires the lock.
The reader (2717) reads the value 4
The last reader releases the lock.
The first reader acquires the lock.
The reader (2716) reads the value 4
The last reader releases the lock.
The first reader acquires the lock.
The reader (2717) reads the value 4
The last reader releases the lock.
The first reader acquires the lock.
The reader (2716) reads the value 4
The last reader releases the lock.
The writer acquires the lock.
The writer (2715) writes the value 5
```

- شرایط مسابقه (race condition) را برای این مساله بدون در نظر گرفتن راه حل سمافور ذکر کنید.
- مشکلات راه حل پترسون چیست؟ یک مثال بیاورید که مشکل این روش را نشان دهد.
- یک مساله ارائه دهید که قابل حل شدن با مانیتور باشد ولی قابل حل شدن با سمافور نباشد (ذکر صورت مساله و دلیل عدم توانایی برای حل مساله با سمافور کافیست و نیازی به نوشتن کد برای این قسمت نیست).

نکته: حل مساله با راه حل مانیتور نمره امتیازی دارد (مساله با سمافور هم باید پیاده سازی شده باشد).

Talk is cheap. Show me the code.
“Linus Torvalds”