

گزارش دستور کار هفتم آزمایشگاه سیستم‌های عامل

نگار موقتیان، ۹۸۳۱۰۶۲

پیاده‌سازی الگوریتم بانکداران

در این آزمایش می‌خواهیم الگوریتم بانکداران را پیاده‌سازی کرده و با استفاده از آن درخواست‌های چند مشتری را مدیریت کنیم. الگوریتم بانکداران یک الگوریتم است که محافظه‌کارانه بررسی می‌کند با اجابت درخواست هر مشتری سیستم در وضعیت امن قرار خواهد داشت یا خیر. در صورتی که سیستم در شرایط امن قرار داشت درخواست اجابت شده و در غیر این صورت رد می‌شود.

* به دلیل طولانی بودن کد این بخش از آزمایش، کد به صورت مستقیم در اینجا آورده نمی‌شود و تنها توضیحات مربوط به آن داده می‌شود.

در ابتدای برنامه کتابخانه‌های مورد نیاز اضافه شده‌اند.

سپس تعداد منابع و مشتری‌ها تعریف شده‌اند (مطابق دستور کار با مقدار ۵).

در ادامه یک mutex lock از جنس pthread_mutex_t تعریف کرده‌ایم تا پیش از ورود به نواحی بحرانی آن را قفل کرده و پس از خروج از آن قفل را آزاد کنیم. در این برنامه نواحی بحرانی نواحی‌ای هستند که با جداول مشترک maximum، available، allocation و need کار می‌کنیم. پس از آن جداول maximum، available، allocation و need مطابق کاربریشان با سایز مناسب تعریف شده‌اند.

تابع printRequest صرفاً برای چاپ کردن یک آرایه از request ها استفاده می‌شود.

تابع checkSafety مطابق شبه‌کد موجود در دستور کار براساس وضعیت فعلی سیستم نوشته شده‌است.

تابع request_resources نیز مطابق شبه‌کد موجود در دستور کار نوشته شده‌است و بررسی می‌کند که آیا درخواست مشتری می‌تواند اجابت شود یا خیر. درخواست تنها در صورتی اجابت می‌شود که با قبول آن سیستم در وضعیت امن باقی بماند.

تابع `release_resources` برای آزاد کردن منابع به کار می‌رود و در این برنامه این کار صرفاً با آپدیت کردن جداول وضعیت سیستم انجام می‌شود.

تابع `runCustomer` تابعی است که هر یک از مشتریان آن را اجرا می‌کند. در این تابع به طور مداوم هر یک از مشتریان به طور تصادفی تعدادی منبع از منابعی که در دست دارد را آزاد کرده (با استفاده از تابع `release_resources`) و یا تعدادی منبع را درخواست می‌کند (با استفاده از تابع `request_resources`).

حال می‌توانیم برنامه مورد نظر را اجرا کنیم. در مثال بررسی شده، ماتریس `max` به صورت زیر مقداردهی اولیه شده است (با توجه به مقادیر نوشته شده در فایل `max.txt`):

```
7 5 3 1 0
3 2 2 3 4
9 0 2 5 5
2 2 2 2 2
4 3 3 2 1
```

همان طور که در شکل زیر دیده می‌شود هر یک از `thread` ها به صورت تصادفی برای بدست گیری منابع یک درخواست داده و یا تعدادی منبع را آزاد می‌کند. با استفاده از این الگوریتم می‌توانیم مطمئن باشیم این درخواست‌ها می‌توانند تا ابد تکرار شوند بدون اینکه حالت `deadlock` رخ دهد

```
maxeu@ubuntu: ~/Desktop/OS Lab/Lab07
File Edit View Search Terminal Help
maxeu@ubuntu:~/Desktop/OS Lab/Lab07$ gcc OSLab07_Bankers\ Algorithm.c -o bankers -pthread
maxeu@ubuntu:~/Desktop/OS Lab/Lab07$ ./bankers 12 8 7 9 7
[RELEASED] - Customer number 0 - [0, 0, 0, 0, 0]
[DECLINED] - Customer number 4 - [4, 3, 3, 1, 1]
[RELEASED] - Customer number 2 - [0, 0, 0, 0, 0]
[RELEASED] - Customer number 1 - [0, 0, 0, 0, 0]
[DECLINED] - Customer number 3 - [0, 1, 1, 2, 1]
[RELEASED] - Customer number 1 - [0, 0, 0, 0, 0]
[RELEASED] - Customer number 3 - [0, 0, 0, 0, 0]
[RELEASED] - Customer number 3 - [0, 0, 0, 0, 0]
[RELEASED] - Customer number 0 - [0, 0, 0, 0, 0]
[DECLINED] - Customer number 4 - [2, 3, 2, 1, 0]
[ACCEPTED] - Customer number 2 - [8, 0, 1, 3, 0]
[RELEASED] - Customer number 1 - [0, 0, 0, 0, 0]
[ACCEPTED] - Customer number 2 - [0, 0, 1, 1, 1]
[ACCEPTED] - Customer number 1 - [1, 1, 1, 1, 1]
[ACCEPTED] - Customer number 3 - [0, 1, 2, 0, 2]
[RELEASED] - Customer number 0 - [0, 0, 0, 0, 0]
[ACCEPTED] - Customer number 2 - [0, 0, 0, 0, 2]
[ACCEPTED] - Customer number 4 - [1, 1, 0, 1, 1]
[RELEASED] - Customer number 1 - [0, 0, 1, 0, 1]
[RELEASED] - Customer number 2 - [3, 0, 1, 0, 0]
[ACCEPTED] - Customer number 0 - [0, 0, 1, 0, 0]
[RELEASED] - Customer number 3 - [0, 1, 0, 0, 2]
[ACCEPTED] - Customer number 4 - [1, 0, 3, 0, 0]
[RELEASED] - Customer number 4 - [2, 1, 2, 1, 0]
[ACCEPTED] - Customer number 1 - [0, 0, 0, 0, 2]
[RELEASED] - Customer number 1 - [0, 0, 0, 0, 1]
[ACCEPTED] - Customer number 2 - [3, 0, 1, 1, 1]
[ACCEPTED] - Customer number 3 - [2, 0, 0, 2, 0]
[ACCEPTED] - Customer number 2 - [1, 0, 0, 0, 0]
[RELEASED] - Customer number 2 - [3, 0, 0, 5, 4]
[DECLINED] - Customer number 0 - [1, 1, 0, 1, 0]
[DECLINED] - Customer number 3 - [0, 1, 0, 0, 0]
[RELEASED] - Customer number 0 - [0, 0, 1, 0, 0]
[ACCEPTED] - Customer number 4 - [1, 2, 1, 0, 0]
```