**گزارش دستورکار هفتم آزمایشگاه سیستم­های عامل**

نگار موقتیان، 9831062

**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 دستورکار با مقدار 5).

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 available، maximum، allocation و need کار می­کنیم. پس از آن جداول available، maximum، allocation و need مطابق کاربردشان با سایز مناسب تعریف شده­اند.

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

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

تابع request\_resources نیز مطابق شبه­کد موجود در دستورکار 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 صرفا 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 تصادفی 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 برنامۀ مورد نظر را اجرا کنیم. در مثال بررسی شده، ماتریس max به صورت زیر مقداردهی اولیه شده­است (با توجه به مقادیر نوشته شده در فایل max.txt):

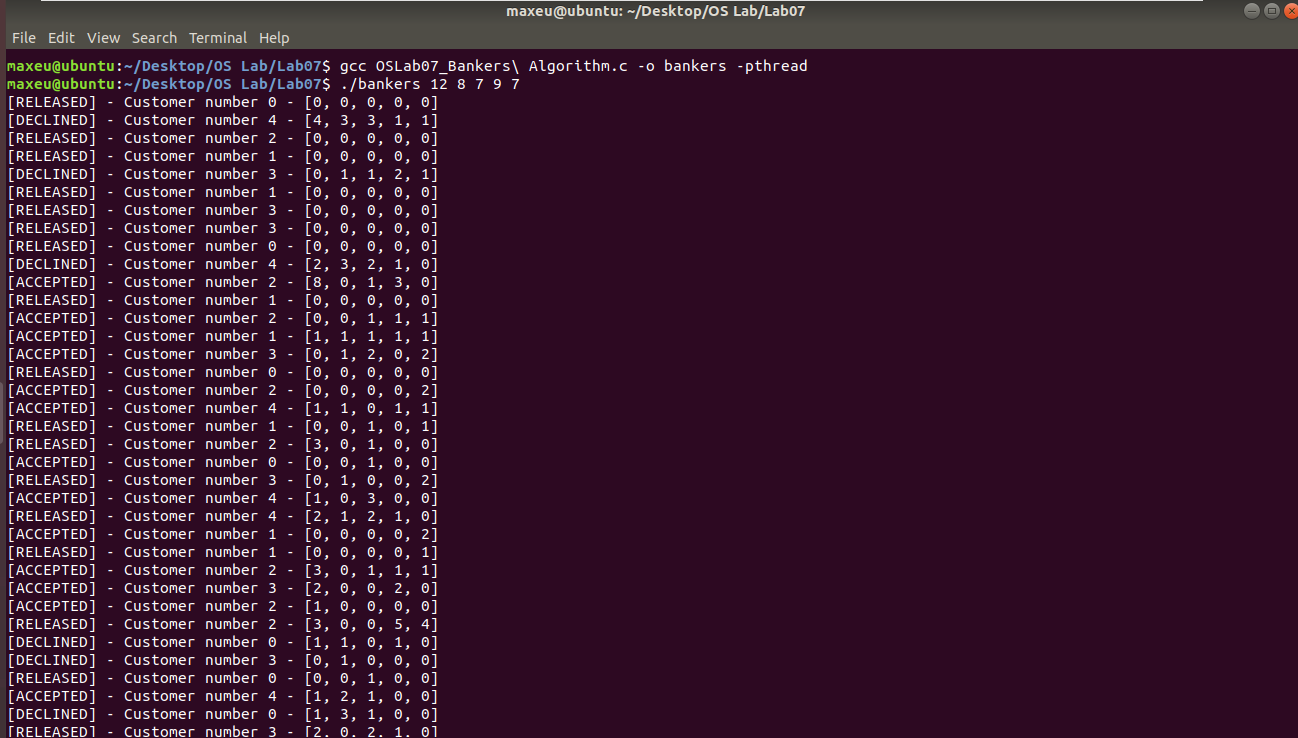
7 5 3 1 0

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