آزمایشگاه سیستم عامل

دستورکار ۱۴: الگوریتم بانکداران

الگوریتم بانکدار یک الگوریتم تخصیص منابع و اجتناب از بن‌بست است. در این الگوریتم مشتری‌های متعددی منابع را از بانک درخواست می‌کنند و سپس بعد از استفاده آن را بازمی‌گردانند. بانکدار تنها در صورتی که در ازای یک درخواست، منبع را اعطا خواهد کرد که سیستم در حالت امن باقی بماند و درخواست منبعی که سیستم را در یک حالت ناامن باقی می‌گذارد، رد می‌شود. برخی از منابعی که در سیستم‌های واقعی دنبال می‌شوند حافظه، سمافورها و رابط دسترسی هستند. نام الگوریتم بانکدار برگرفته از این حقیقت است که این الگوریتم می‌تواند در سیستم بانکی مورد استفاده قرار گیرد تا تضمین کند که بانک همه منابع خود را از دست نمی‌دهد. چون بانک هیچ‌گاه پول را به نحوی تخصیص نمی‌دهد که نتواند نیازهای بقیه مشتری‌ها را برطرف کند. با الگوریتم بانکدار بانک تضمین می‌کند که زمانی که مشتری‌ها پول درخواست می‌کنند بانک هیچ‌گاه به حالت ناامن نمی‌رود. اگر درخواست مشتری باعث نشود که بانک حالت امن را ترک کند، منبع اختصاص می‌یابد در غیر این‌صورت مشتری باید صبر کند تا زمانی که مشتری‌های دیگر سپرده منبع کافی قرار دهند.

زمانی یک حالت را امن گوییم که حداقل یک ترتیب از فرایندها وجود دارد که تمام فرایندها می‌توانند تا کامل شدن اجرا شوند. از آن‌جایی که سیستم نمی‌تواند بداند یک فرایند چه زمانی به پایان می‌رسد یا چه تعداد منابع درخواست خواهد شد، پس سیستم فرض می‌کند همه فرایندها تلاش می‌کنند که حداکثر منابع را در اختیار داشته باشند تا زودتر به پایان برسند. از آنجایی که سیستم به‌طور ویژه اهمیت نمی‌دهد که اجرای هر فرایند چقدر طول می‌کشد پس این یک فرض معقول در اکثر موارد است. هم‌چنین اگر یک فرایند بدون دستیابی به حداکثر منابع پایان یابد فقط اجرا را روی سیستم ساده‌تر کرده است. حالت امن در نظر گرفته شده که تصمیم بگیرد فرآیند به صف آماده برود.

شبه‌کد بررسی امن بودن وضعیت به صورت زیر است:

1) Let Work and Finish be vectors of length 'm' and 'n' respectively.
Initialize: Work = Available
Finish[i] = false; for i=1, 2, 3, 4....n

2) Find an i such that both
a) Finish[i] = false
b) $Need_i$ <= Work
if no such i exists goto step (4)

3) Work = Work + Allocation[i]
Finish[i] = true
goto step (2)

4) if Finish [i] = true for all i
then the system is in a safe state

حال فرض کنید $Request_i$ آرایه درخواست‌های فرآیند P_i است. $Request_i[j] = k$ به این معنی است که فرآیند P_i تعداد k تا از منبع R_j درخواست دارد. شبه کد درخواست منبع به صورت زیر است:

1) If $Request_i$ <= $Need_i$
Goto step (2) ; otherwise, raise an error condition, since the process has exceeded its maximum claim.

2) If $Request_i$ <= Available
Goto step (3); otherwise, P_i must wait, since the resources are not available.

3) Have the system pretend to have allocated the requested resources to process Pi by modifying the state as
follows:
Available = Available – Requesti
$Allocation_i$ = $Allocation_i$ + $Request_i$
$Need_i$ = $Need_i$– $Request_i$

بانکدار:

بانکدار درخواست‌های n مشتری را برای m نوع منبع بررسی خواهد کرد. برای سادگی، فرض کنید ۶ نوع منبع داریم. بانکدار با استفاده از ساختمان داده‌های زیر پیگیر منابع خواهد بود:

```
#define NUMBER_OF_RESOURCES 5
/* this maybe any values >= 0 */
#define NUMBER_OF_CUSTOMERS 5
/* the available amount of each resource */
int available[NUMBER_OF_RESOURCES];
/* the maximum demand of each customer*/
int maximum[NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
/* the amount currently allocated to each customer */
int allocation[NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
/* the remaining need of each customer */
int need] NUMBER_OF_CUSTOMERS][NUMBER_OF_RESOURCES];
```

مشتری(فرآیند):

تعداد n مشتری(فرآیند) ایجاد کنید که منابع بانک را درخواست و سپس آزاد کنند. مشتری‌ها(فرآیندها) به صورت مداوم، تعدادی تصادفی از منابع را درخواست و پس خواهند داد. درخواست‌های مشتری‌ها برای منابع به مقادیر متناظر آن‌ها در آرایه need محدود

خواهند بود. بانکدار در صورتی با اعطای یک درخواست موافقت خواهد کرد که شروط مطرح شده در الگوریتم ایمنی بانکداران را ارضا نماید. اگر درخواستی سیستم را در یک حالت امن باقی نگذارد، بانکدار آن را کنار خواهد زد.

تمرین:

الگوریتم بانکداران را با توجه به توضیحات داده شده پیاده سازی کنید. به طوری که تعداد مشتری‌ها (فرآیندها) و انواع منابع و مقادیر کلی آن‌ها را از کاربر دریافت و بر اساس ورودی، حداکثر مقادیر موردنیاز هریک از منابع را دریافت کنید. به ازای هریک از فرآیندها مقادیر منابع مورد نیاز و تخصیص یافته را نیز از کاربر دریافت کنید. سپس با اجرای الگوریتم نتایج تخصیص منابع را مشخص کنید.