درس سیستم‌های عامل (دکتر جلیلی)

زمستان ۱۳۸۶

تمرین کامپیوتری ۴

تاریخ تحویل : شنبه ۱۳۸۶/۱۱/۶

نکات قابل توجه

۱. تمرین‌های کامپیوتری باید در گروه‌های **دو نفره** انجام شود.

۲. برنامه‌ها باید در محیط لینوکس و با زبان C/C++ نوشته شوند.

۳. استفاده از Makefile برای ساخت برنامه‌ها الزامی است. اطلاعات مربوط به نوشتن Makefile در بخش منابع وب‌سایت درس موجود است.

۴. تمارین باید تا پایان وقت تعیین شده، به آدرس ce424-owner@lists.ce.sharif.edu فرستاده شوند.

- عنوان میل : CA4
- ذکر نام و نام خانوادگی و شماره دانشجویی در متن نامه الزامی است.
- متن برنامه به صورت zip شده و فرمت (family1_family2.zip) پیوست شود.

۵. تنها در صورت بروز مشکل در فرستادن تمارین به آدرس فوق، می‌توانید آن را به آدرس ce424s86@gmail.com بفرستید.

۱. الگوریتم Bakery

برنامه‌ای بنویسید که الگوریتم Bakery (برای مسأله‌ی همگام‌سازی بیشتر از ۲ ترد) در کتاب درسی را پیاده‌سازی کند. شما باید یک برنامه‌ی چندرشته‌ای (multi-thread) ساده برای تست برنامه‌ی اصلی خود بنویسید. برای مثال، هر رشته، یک فیلد مشترک را داخل یک while بی‌نهایت Increment می‌کند و روی صفحه‌ی نمایش چاپ می‌کند. استفاده از خود متدهای همگام‌سازی C/C++ مجاز نمی‌باشد.

۲. الگوریتم Bankers

هدف این قسمت، شبیه‌سازی الگوریتم Bankers برای مسأله‌ی بن‌بست می‌باشد. پس از اجرای برنامه، اطلاعات منابع و پردازه‌ها از یک فایل ورودی خوانده می‌شود. فرمت فایل ورودی به همراه یک مثال در زیر آورده شده است.

Input File Format:

```
ProcessesNum ResourcesTypeNum
Resources:
resource1Num resource2Num resource3Num …
Max:
Max matrix
Allocated:
Allocated matrix
```

Sample Input File:

```
3 3
Resource:
9 9 12
Max:
5 7 10
7 3 2
3 3 3
Allocated:
3 6 9
4 1 0
1 1 2
```

توجه شود که PIDها از ۱ شروع می‌شوند (در مثال فوق: ۱و۲و۳).

دستورات

أ- **kill** - منابع یک پردازه را آزاد می‌کند و پردازه را خاتمه می‌دهد. مثال:

kill:

```
/>kill processID
Resources were released:
R1 R2 R3
3  6  9
/>
```

ب- **req** - با این دستور، هر پردازه می‌تواند یک سری منابع را درخواست دهد و یا آزاد کند. (علامت مثبت به معنای درخواست و علامت منفی به معنای آزاد کردن منابع است). این دستور، پس از گرفتن PID پردازه، تعداد منابع درخواستی یا آزادشده به ازای آن PID را از ورودی می‌گیرد و خروجی‌های مختلفی ممکن است تولید کند. مثال:

req:

```
/>req 2
request: 2 0 2
System has not enough resources for this
request.

/>req 2
request: 2 0 1
This request could cause a deadlock.

/>req 3
request: 1 0 1
The request was accepted.

/>req 1
request: -2 -2 -2
The request was accepted.
```

برای نمونه، مورد آخر به این معناست که پردازه‌ی ۱، ۲ واحد از هر منبع اختصاص‌یافته را آزاد می‌کند.

ت- **safeseq** - یک حالت امن، برای اجرای کامل تمام پردازه‌ها نمایش می‌دهد (در صورت وجود).

ث- **showmaxneed** – ماتریس حداکثر نیاز را نشان می‌دهد.

ج- **showallocation** – ماتریس منابع اختصاص‌یافته در حال حاضر را نشان می‌دهد.

ح- **quit** – از برنامه خارج می‌شود.