

## پروژه پایانی درس سیستم عامل

در این پروژه می‌خواهیم ماژول زمانبندی برای پردازنده های multi-core طراحی کنیم. فرض کنید در ابتدا ما M عدد process داریم. هر process ما دارای arrival time و CPU burst و Core-needed هستند. با دو مفهوم اول در تمرینات آشنا شدیم. Core-needed به معنای تعداد هسته های لازم از cpu است که این process برای انجام کار خود به آنها نیاز دارد. ما برای زمانبندی یک cpu که دارای N هسته است در اختیار داریم و هر هسته توانایی انجام یک کار را دارد (یعنی فقط در اختیار یک process قرار میگیرد). برای مثال یک process برای انجام کار خود نیازمند یک هسته است ولی process دیگر نیازمند 2 هسته است.

### ماژول زمانبندی :

این ماژول باید در هر لحظه بر اساس الگوریتمی که شما مینویسید و مبنای آن براساس الگوریتم FCFS است هر process را که در آن لحظه در وضعیت ready قرار دارد ، زمانبندی کند. توجه کنید که ماژول زمانبندی ما میتواند در هر لحظه cpu را در اختیار چند process بگذارد. همچنین این ماژول وظیفه ی درخواست هسته های لازم برای هر process بر عهده دارد.

- طبیعتا الگوریتم به صورت قبضه ای نیست.

### فرآیند ها :

هر process توسط یک کلاس تعریف میشود و طول موجودیت خود به تعداد تصادفی و در فواصل زمانی تصادفی به ماژول زمانبندی درخواست میدهد. هر فراخوانی ماژول زمانبندی را به شکل موازی و با یک thread جداگانه اجرا میکند ( پیشنهاد میشود از ThreadPool استفاده کنید). زمانی که کار process تمام شد باید هسته های اختصاص داده شده به آن process آزاد شود. اگر فرآیندی در یک لحظه بدلیل مشغول بودن همه هسته های پردازنده قابل زمانبندی نبود باید صبر کند تا هسته های مورد نیازش آزاد شوند.

### پردازنده :

کلاس پردازنده شامل تعداد هسته های آن است. این کلاس دارای یک تابع برای اختصاص دادن هسته های cpu است. توجه کنید که دسترسی به هسته یک نقطه ی بحرانی محسوب میشود و باید توسط lock دسترسی به هسته های پردازنده را مدیریت کرد.

- تعداد هسته های مورد نیاز هر فرآیند کوچکتر از N است.
- انتخاب نوع lock بر عهده ی خودتان است.
- از ساختمان داده هایی که خودشان thread-safe هستند و در زبان جاوا در کتابخانه java.util.concurrent وجود دارند استفاده نکنید. در استفاده از ساختمان داده ها باید خود شما کار محافظت از داده را انجام دهید.

## نکات مهم :

- اگر در یک لحظه تمام هسته های پردازنده مشغول باشند و الگوریتم شما نتواند آنرا زمانبندی کند باید خطای مشغول بودن پردازنده تولید شود.
- هر 5 ثانیه یک thread وضعیت سیستم را گزارش میدهد که شامل :
  - تعداد هسته های آزاد
  - تعداد هسته های مشغول به کار
  - مدت زمان اجرا، فرآیند های در حال اجرا ، فرآیند های در صف اجرا و ...

## بخش امتیازی :

- الگوریتم خود را براساس priority queue طراحی کنید.
- طراحی رابط کاربری به صورتی که گزارش سیستم در آن قابل مشاهده باشد.
- پیاده سازی وقفه : با وقفه ها و نحوه ی کارکرد آنها در درس آشنا شدیم. در این پروژه برای پیاده سازی این سیستم باید یک سری از فرآیند های ما اولویت بیشتری نسبت به بقیه داشته باشند و وقتی زمان اجزای آنها فرا می رسد باید تمام فرآیند های دیگر متوقف شوند و state سیستم ذخیره شود. پس از اینکه کار این فرآیند به اتمام رسید سیستم به حالت قبلی خود باز میگردد و فرآیندهایی که قبلا در حال اجرا بودند به کار خود ادامه میدهند. توجه کنید که هنگامی که وقفه رخ میدهد ممکن است فرآیندهای دیگری هم در وضعیت آماده قرار بگیرند و وظیفه الگوریتم شماست که بعد از پایان یافتن وقفه آن ها را مدیریت کند.