به نام هستی بخش

سیستم های عامل

نیم‌سال اول 1403-1404

دانشکده‌ی مهندسی و علوم کامپیوتر

پروژه پایانی

مدرس: دکتر ابراهیمی مقدم

تاریخ تحویل: ۱۱ بهمن

در این پروژه قصد داریم یک سیستم زمان‌بندی (Scheduling) چندپردازنده‌ای برای اجرای فرایندهای (Processes) ورودی پیاده‌سازی کنیم. نکته‌ی کلیدی در این پروژه آن است که برخلاف پروژه‌های رایج که ورودی از طریق یک فایل مشخص خوانده می‌شود، در اینجا فرایندها به صورت پویا و اتفاقی (Random) در حین اجرای برنامه تولید می‌گردند. هر فرایند دارای زمان ورود، محدودیت زمانی برای شروع به اجرا (حداکثر زمان شروع)، مدت زمان اجرای مورد نیاز، و یک امتیاز است. سیستم باید به گونه‌ای عمل کند که حداکثر امتیاز کلی حاصل شود و در عین حال محدودیت‌ها (مانند ددلاین یا حداکثر زمان شروع) رعایت گردد.

ویژگی‌های کلی سیستم

1. **تولید تصادفی فرایندها**
   - به جای دریافت فرایندها از طریق فایل، در این پروژه یک مولد فرایند (Process Generator) وجود دارد که به صورت همزمان (در قالب یک ترد جداگانه) فرایندها را تولید می‌کند.
   - هر فرایند در لحظه‌ی ورود، مجموعه‌ای از ویژگی‌ها را به خود اختصاص می‌دهد (که همگی به صورت رندوم تعیین می‌شوند - به جز Arrival Time):
     - **زمان ورود (Arrival Time)**: زمانی که فرایند تولید و وارد سیستم می‌شود.
     - **مدت زمان اجرای مورد نیاز (Execution Time)** یا حداکثر زمانی که CPU باید به آن اختصاص یابد.
     - **ددلاین شروع فرایند (Starting Deadline):** حداکثر زمانی که یک فرا برای گرفتن CPU می‌تواند منتظر بماند.
     - **ددلاین پایان فرایند (Ending Deadline)** یا حداکثر زمانی که شروع اجرای فرایند باید قبل از آن صورت گرفته باشد (یا بسته به نوع پیاده‌سازی، کامل‌شدن فرایند تا قبل از رسیدن به ددلاین).

- **امتیاز (Value/Score)** که عددی بین ۱ تا ۱۰۰ به صورت تصادفی به فرایند داده می‌شود.
    - می‌توان فرض کرد در هر لحظه ممکن است فرایند جدیدی وارد شود و تا زمانی که ۱۰۰ فرایند به صورت کامل اجرا شده باشند، فرایندها تولید می‌شوند.

2. **مدیریت فرایندها**
    - فرایند تولید شده ابتدا وارد یک صف (یا ساختار داده‌ای که انتخاب می‌کنید) می‌شود. این صف می‌تواند به عنوان صف ورودی (Input Queue) در نظر گرفته شود.
    - سپس اسکجولر می‌تواند به صورت پویا فرایندها را از این صف برداشته و وارد لیست (یا صف) آماده (Ready Queue یا همان لیست اسکجولر) کند.
    - ممکن است در طول زمان، اسکجولر برخی فرایندها را از صف آماده خارج کرده و دوباره به صف ورودی برگرداند (مثلاً وقتی تصمیم می‌گیرد فرایند ارزشمندی وارد شود و ظرفیت محدود باشد). در نتیجه امکان جابه‌جایی بین این دو صف وجود دارد.
    - هر فرایند اگر تا زمان ددلاین خودش شروع به اجرا نشده باشد، دیگر ارزشی برای سیستم ندارد و باید حذف شود.
3. **چند CPU و اجرای فرایند**
    - سیستم دارای چند CPU (مثلاً ۳ CPU) است. برای ساده‌سازی، هر CPU را به صورت یک ترد (Thread) مجزا پیاده‌سازی کنید.
    - هر یک از CPUها، هنگامی که آزاد است، به سراغ صف (یا لیست) آماده می‌رود و یکی از فرایندهای موجود را برای اجرا برمی‌دارد.
    - انتخاب اینکه کدام فرایند توسط CPU برداشته شود، می‌تواند بر اساس استراتژی‌های گوناگون (مثلاً کوتاه‌ترین زمان اجرای باقی‌مانده، بیشترین امتیاز، کمترین ددلاین، و ...) باشد. این انتخاب در CPU صورت می‌گیرد. باید هم‌زمانی (Concurrency) رعایت شود تا CPUها در دسترسی به این منبع مشترک (صف فرایندهای آماده) دچار Race Condition نشوند.
    - زمانی که یک CPU مشغول اجرای یک فرایند است، در آن بازه زمانی دیگر CPUها نیز می‌توانند در صورت تمایل، فرایندهای دیگری را بردارند و اجرا کنند، تا زمانی که صف آماده خالی شود یا همه CPUها مشغول باشند.
4. **امتیازدهی و بهینه‌سازی**
    - هر فرایند دارای یک امتیاز مشخص است که در هنگام تولید تصادفی تعیین می‌شود.
    - هدف نهایی اسکجولر، حداکثر کردن مجموع امتیاز فرایندهای اجرا شده است. فرایندی که به موقع اجرا شده باشد (یعنی بتوانیم قبل از رسیدن به ددلاین شروع آن را شروع و قبل از رسیدن به ددلاین پایانی آن را تمام کنیم)، امتیازش به سیستم اضافه می‌شود. اگر فرایند از دست برود (Miss)، آن امتیاز را از دست می‌دهیم.

- طراحی و پیاده‌سازی الگوریتم زمان‌بندی برعهده‌ی شماست و می‌توانید از الگوریتم‌های شناخته‌شده (Deadline-based, Priority-based, ...) استفاده کنید یا الگوریتم ابتکاری خود را پیاده‌سازی نمایید (در صورت پیاده‌سازی الگوریتم ابتکاری ۵ درصد نمره امتیازی به شما تعلق می‌گیرد).

نیازمندی‌ها و مراحل پیاده‌سازی

1. تولید فرایند (Process Generator)
   - یک ماژول یا ترد که در فواصل زمانی مختلف، فرایندهای جدیدی تولید می‌کند.
   - برای هر فرایند ویژگی‌های ذکر شده (مدت اجرا، ددلاین، امتیاز و ...) را تعیین نموده و آن را وارد صف ورودی می‌کند.
2. مدیریت صف‌ها و اسکجولر (Scheduler)
   - صف ورودی: محل ورود اولیه فرایندهای تولید شده توسط مولد که ظرفیت آن نامحدود است.
   - صف یا لیست آماده (Ready Queue): محلی که اسکجولر فرایندها را برای انتخاب توسط CPUها در آن قرار می‌دهد.
   - اسکجولر باید در هر لحظه، تصمیم بگیرد که کدام فرایندها از صف ورودی به صف آماده بروند. همچنین امکان دارد برخی فرایندها در صف آماده جای خود را به فرایندهای تازه‌وارد با ارزش بالاتر بدهند.
   - اندازه لیست آماده ۲۰ در نظر گرفته شود و در صورت پر شدن آن، اسکجولر باید تصمیم بگیرد که آیا فرایند جدید ارزشمندتری را جایگزین فرایند دیگری بکند یا خیر.
3. اجرای فرایندها توسط CPUها
   - برای هر CPU یک ترد مجزا در نظر بگیرید. این تردها به صورت همزمان فعال هستند.
   - هنگامی که یک CPU آزاد می‌شود، به سراغ صف آماده رفته و با یک مکانیزم امن (مثلا Mutex یا Semaphore) فرایندی را انتخاب و حذف می‌کند تا آن را اجرا نماید.
   - در طول اجرای یک فرایند، CPU تا اتمام آن، فرایند دیگری را انتخاب نمی‌کند.
   - پس از اتمام اجرای فرایند، امتیاز مربوطه به سیستم اضافه می‌گردد و فرایند از سیستم خارج می‌شود.
4. همزمانی (Concurrency) و منبع مشترک
   - باید مکانیزم‌های همزمانی به کار گرفته شوند تا وقتی چند CPU به طور همزمان قصد دسترسی به صف آماده را دارند، دچار مشکلات Race Condition نشوید.

نکات تکمیلی

1. **آزادی در انتخاب الگوریتم**
   - می‌توانید از الگوریتم‌های استاندارد یا ترکیبی از آن‌ها استفاده کنید.
   - الگوریتم ابتکاری (Heuristic) یا هر نوع ایده خلاقانه برای مدیریت ددلاین و امتیاز فرایند نیز قابل قبول است.
2. **طراحی چندبخشی**
   - پروژه را می‌توان در سه بخش کلی دید:
     1. **تولیدکننده‌ی فرایند (Process Generator)** که مسئول تولید تصادفی فرایندهاست.
     2. **اسکجولر (Scheduler)** که فرایندها را بین صف ورودی و صف آماده مدیریت می‌کند.
     3. **تردهای CPU** که فرایندها را از صف آماده دریافت و اجرا می‌کنند.
3. **مدیریت منبع مشترک**
   - در بخش امتیازی پیشنهاد می‌شود که CPUها به‌طور مستقل و رندوم، یا بر اساس معیار خودشان، فرایندها را از صف آماده بردارند. این کار نیازمند هم‌زمانی سطح پایین‌تر است و باید مکانیزم‌های هم‌زمانی پیاده‌سازی شود تا از ناسازگاری جلوگیری شود.
4. **پیاده‌سازی Process Generator**
   - دقت شود در مواردی مثل تخصیص ویژگی‌های هر فرایند که خواسته شده مقادیر به صورت رندوم تعیین گردند، رنج اعداد رندوم تولیدی، باید به شکل مناسبی توسط خودتان تعیین گردند به طوری که مثلا Starting Deadline در Arrival Time صفر می‌تواند بین ۰ تا ۵ باشد و با گذشت زمان این بازه Starting Deadline تغییر کند (مثلا به ۳ تا ۸).
   - در شروع هر بازه زمانی تعداد Process های جنریت شده حداکثر برابر ۱۰ می‌باشد.
5. **محدودیت‌ها**
   - در این پروژه تنها مجاز به استفاده از کتابخانه‌ی استاندارد زبان برنامه‌نویسی خود هستید و حق استفاده از کابخانه‌های دیگر را ندارید.
   - برای پیاده‌سازی تنها مجاز به ساختارهای ساده‌ی اجرای همروند هستید: pthread و std::thread در زبان سی و سی‌پلاس‌پلاس و کلاس Thread و interface Runnable در زبان جاوا و گو روتین در زبان گولنگ.
   - برای پیاده‌سازی تنها مجاز به استفاده از ساختارهای ساده‌ی کنترل همروندی از جمله mutex (فقط از نوع ساده و نه read-write lock) و semaphore هستید و امکانات دیگر مثل ساختمان‌داده‌های تردسیف باید برای استفاده خودتان پیاده‌سازی شوند. در زبان گو امکان استفاده از چنل‌ها را نیز **ندارید**.

بخش امتیازی

1. پردازش چندباره و استخراج آمار
   - برنامه را بارها (مثلاً ۱۰ یا ۲۰ بار) اجرا کنید. هر بار یا تا زمانی که تعداد مشخصی فرایند تولید شود (مثلاً ۵۰ فرایند) یا تا رسیدن به زمانی خاص (مثلاً زمان ۲۰) آن را پیش ببرید.
   - در هر بار اجرای برنامه، مقادیر HIT RATE، مجموع امتیاز، تعداد فرایندهای منقضی‌شده و سایر شاخص‌های مهم را ذخیره کنید.
   - در انتها نتایج این اجراها را با یکدیگر مقایسه کرده و به‌صورت نمودارها یا جدول‌های آماری ارائه دهید.

خروجی و گزارش نهایی

- در پایان اجرای برنامه (پس از اتمام تعداد مشخصی فرایند) موارد زیر باید نمایش داده شود:
  1. لیست فرایندهایی که با موفقیت اجرا شده‌اند، همراه با زمان شروع و پایان اجرای آنها.
  2. امتیاز اختصاصی هر فرایند اجرا شده و مجموع امتیاز سیستم.
  3. تعداد فرایندهایی که از دست رفته‌اند (به دلیل منقضی شدن زمان شروع یا رسیدن ددلاین).
  4. میانگین زمان انتظار، میانگین زمان پاسخ یا سایر معیارهای دلخواه.
- در صورت انجام بخش امتیازی، نمودار یا جدول‌های آماری مقایسه‌ای نیز ارائه شود.

نکات پایانی

1. سوالات خود را در دیسکاشن کانال مطرح کنید یا به آیدی @sadramousavi77 و @Amirhossein_Sadr پیام بدهید.
2. فرمت نامگذاری پروژه به صورت OS-PROJECT-[Student ID]-[Student Name] باشد.
3. در صورت مشاهده هرگونه تقلب، نمره صفر برای افراد خاطی لحاظ خواهد شد.

موفق باشید