

پروژه زمانبندی

شرح پروژه

قصد داریم در این پروژه یک scheduler را برای یک سیستم با چهار پردازنده پیاده سازی کنیم. در شبیه ساز ما یک ترد برای هر پردازنده استفاده می شود. همچنین از یک ترد جداگانه که با چهار پردازنده در ارتباط است استفاده می شود. وظیفه این ترد چاپ خروجی برنامه است که شامل اطلاعات مربوط به تسک انجام شده و پردازنده ای که آن را انجام داده است. در سیستم ما منابعی وجود دارد که برای انجام تسک ها به آن ها نیاز است. تعداد این منابع محدود است و تسک های مختلف به مجموعه منابع متفاوتی نیاز دارند. هنگامی که یک تسک در اول صف اولویت قرار می گیرد ولی منابع مورد نیاز برای آن موجود نباشد از صف ready خارج شده و وارد صف waiting می شود. در شبیه ساز باید برای این موضوع راه حلی پیشنهاد شود.

منابع

در سیستم ۳ نوع منبع A,B,C وجود دارد که هنگام شروع برنامه تعداد موجود از هر کدام در سیستم به شما داده می شود.

تسک

در این سیستم ۳ نوع تسک X,Y,Z وجود دارد. اولویت تسک ها به ترتیب برای Z از همه بیشتر و معادل 1 بعد آن برای Y معادل 2 و در نهایت X که معادل 3 است.

- تسک X به منابع A و B نیاز دارد.

- تسک Y به منابع B و C نیاز دارد.

- تسک Z به منابع A و C نیاز دارد.

برای هر تسک مدت زمان مورد نیاز برای اجرای آن تسک نیز به شما داده می شود. همچنین شما باید برای ساختار تسک فیلدی برای ذخیره وضعیت تسک در نظر بگیرید که نشان دهنده استیت آن در سیستم است. (ready-waiting-running) همچنین باید فیلدی مربوط به میزان زمانی که تسک بر روی پردازنده قرار گرفته است تعریف شود.

الگوریتم های زمانبندی

الگوریتم هایی که باید پیاده سازی شوند شامل :

- shortest-Job-First
- First-Come, First-Served
- Round-Robin

موارد نمره اضافه:

- Virtual round robin
- Multilevel feedback queue
- HRRN (Highest Response Ratio next
- Multilevel queue (based on different priorities)

- سه تیمی که بهترین پیاده سازی بخش زمانبندی را ارائه دهند نمره اضافه مازاد بر این ۴ حالت تعلق خواهد گرفت.

صف Ready

این صف مربوط به تسک هایی می شوند که آماده اجرا هستند و ترتیب آن ها با توجه به الگوریتم های زمانبندی مطرح شده مشخص می گردد. همچنین فقط یک صف اولویت در سیستم وجود دارد.

صف waiting

این صف مربوط به تسک هایی می شود که امکان اجرای آن ها است ولی منابع مورد نیاز آن ها موجود نیست. مثلاً هنگامی که یک پردازنده تسکی از صف اولویت اولویت انتخاب می کند ولی منابع آن ها موجود نباشد سیستم این تسک را از صف اولویت خارج کرده و در صف انتظار قرار می دهد. لازم به ذکر است برای جلوگیری از starvation باید راه حلی برای برگرداندن تسک ها به صف اولویت پیاده سازی شود. پس لازم است الگوریتمی برای مرتب کردن این صف با توجه به منابع آزاد سیستم و بدست آوردن بهترین بهره وری از پردازنده ها پیاده سازی شود. نکته قابل توجه دیگر این است که در صورتی که تسکی از صف انتظار به صف اولویت برگردد و مدت زیادی در صف انتظار قرار گرفته بوده یا زمان باقی مانده اجرای آن نسبت به بقیه تسک های بر روی پردازنده ها کم باشد با توجه به الگوریتم زمانبندی یا باید اولویت آن افزایش یابد یا در اول صف قرار بگیرد یا جایش با یکی از تسک های در حال اجرا عوض شود.

زمان

هر واحد زمان را می‌توانید یک لوپ در حلقه اصلی برنامه تان در نظر بگیرید.

سپینک

در شبیه ساز فقط یک صف ready و waiting وجود دارد پس باید از race condition برای ترد های پردازنده هنگام دسترسی به آن ها جلوگیری کرد. برای این کار باید از mutex استفاده کرد. همچنین برای سایر منابع مشترک در سیستم مانند منابع مشترک بین ترد کنترل که وظیفه چاپ وضعیت کلی سیستم و پردازنده ها و تسک های در حال اجرا و صف ها و ترد های پردازنده در صورت امکان رخداد race condition از mutex استفاده کرد

فرمت ورودی خروجی

● ورودی :

- در خط اول به ترتیب از چپ به راست تعداد منابع موجود در سیستم برای A,B,C قرار می گیرد
- در خط بعدی تعداد تسک هایی که قرار است زمانبندی شوند وارد می شود.
- از خط سوم به بعد تسک ها به فرمت TaskName TaskType TaskDuration به عنوان مثال : T2 Y 3

● خروجی :

- بعد از هر واحد زمان وضعیت دو صف موجود در سیستم، تعداد منابع موجود و وضعیت هر پردازنده که شامل تسکی است که در حال انجام است باید نمایش داده شود. همچنین در صورتی که امکان قرار گیری تسکی بر روی پردازنده موجود نبود رو به روی آن Idle به معنی بیکار بودن قرار می گیرد و هنگام تمام شدن تسک ها زمان بیکاری هر یک از پردازنده ها باید چاپ شود. به مثال های زیر توجه کنید :

```
1 A:0 B:1 C:0
2
3 priority queue : T2-T4-T7
4
5 wating queue : T4
6
7 cpu1:T1
8 cpu2:T3
9 cpu3:T6
10 cpu4:T5
```

```
1 A:0 B:1 C:0
2
3 priority queue : T2-T4-T7
4
5 wating queue : T4-T5
6
7 cpu1:T1
8 cpu2:T3
9 cpu3:T6
10 cpu4:Idle
```

```
cpu1 Idle-Time:2
cpu2 Idle-Time:1
cpu3 Idle-Time:0
cpu4 Idle-Time:3
```

توضیحات تکمیلی و نحوه تحویل

- پروژه به صورت گروه های ۲ نفره قابل انجام است.
- استفاده از زبان های پایتون - سی/سی پلاس پلاس - جاوا مجاز است.
- در صورت استفاده از زبان سی/سی پلاس پلاس باید از makefile استفاده شود.
- در یک فایل README مراحل پیاده سازی به طور خلاصه توضیح داده شود.
- کد برنامه و فایل README در قالب یک فایل به صورت P2_Name_studentID.zip ارسال شود

موفق باشید

تیم حل پروژه سیستم عامل