

پروژه دوم درس سیستم عامل

Scheduler

شرح پروژه

در این پروژه از شما خواسته شده است تا یک scheduler را برای سیستمی با چهار پردازنده پیاده سازی کنید. در این شبیه ساز از یک نخ برای هر پردازنده و یک نخ جداگانه که با چهار پردازنده در ارتباط است استفاده می شود. وظیفه نخ مشترک چاپ خروجی برنامه است که شامل اطلاعات مربوط به وظیفه انجام شده و پردازنده ای که آن را انجام داده، می شود.

در سیستم همچنین منابعی (resources) وجود دارد که برای انجام وظیفه ها به آنها نیاز داریم. تعداد این منابع محدود است و وظیفه های مختلف بسته به نوعشان به منابع متفاوتی نیاز دارند. اگر یک وظیفه در اول صف اولویت قرار گیرد ولی منابع مورد نیاز برای آن موجود نباشد، از صف ready خارج شده و وارد صف waiting می شود.

مفاهیم

منابع

در سیستم ۳ نوع منبع (resource) R1, R2, R3 وجود دارد که هنگام شروع برنامه تعداد موجود از هر کدام در سیستم به شما داده می شود.

وظایف

در این سیستم ۳ نوع وظیفه (task) X, Y, Z وجود دارد. اولویت وظیفه ها به ترتیب برای Z از همه بیشتر (=1) بعد از آن برای Y معادل ۲ و در نهایت X معادل ۳ است. نوع و تعداد منابع مورد نیاز این وظایف ثابت و به شرح زیر است:

- وظیفه X به منابع R1 و R2 نیاز دارد.
- وظیفه Y به منابع R2 و R3 نیاز دارد.
- وظیفه Z به منابع R1 و R3 نیاز دارد.

برای هر وظیفه مدت زمان مورد نیاز برای اجرای آن داده می شود. همچنین باید در ساختار وظیفه فیلدی برای ذخیره وضعیت وظیفه در نظر بگیرید که نشان دهنده state آن در سیستم است (ready/waiting/running). فیلدی مربوط به مدت زمانی که از قرار گرفتن وظیفه بر روی گذشته است نیز تعریف شود.

الگوریتم های زمانبندی

الگوریتم هایی که باید پیاده سازی شوند:

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

پیاده سازی الگوریتم های زیر نمره اضافه دارد:

- Multilevel feedback queue
- HRRN

صف Ready

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

صف Waiting

این صف مربوط به وظیفه هایی می شود که امکان اجرای آن ها وجود دارد، ولی منابع مورد نیاز آن ها موجود نیست. مثلاً هنگامی که یک پردازنده وظیفه ای از صف اولویت انتخاب می کند ولی منابع آن موجود نیست، سیستم این وظیفه را از صف اولویت خارج کرده و در صف انتظار قرار می دهد.

برای جلوگیری از starvation، باید راه حلی برای برگرداندن وظیفه ها به صف اولویت در نظر گرفته شود. در نتیجه لازم است الگوریتمی برای مرتب کردن این صف با توجه به منابع آزاد سیستم و به دست آوردن بهترین بهره وری از پردازنده ها پیاده سازی شود.

در صورتی که وظیفه ای 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 وظیفه های 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 وظیفه های در حال اجرا عوض شود.

زمان

هر واحد زمان را می‌توانید یک دور در حلقه اصلی برنامه‌تان در نظر بگیرید.

همگام‌سازی (Synchronisation)

در شبیه‌ساز فقط یک صف ready و waiting وجود دارد (در میان چهار پردازنده مشترک هستند) پس باید از وقوع race condition برای نخ‌های پردازنده هنگام دسترسی به آن‌ها جلوگیری کرد. برای این کار از mutex استفاده کنید.

برای سایر منابع مشترک در سیستم نیز (مانند منابع مشترک بین نخ کنترل که وظیفه چاپ وضعیت کلی سیستم را دارند، پردازنده‌ها، وظیفه‌های در حال اجرا، صف‌ها و نخ‌های پردازنده) در صورت امکان رخداد race condition از mutex استفاده کنید.

فرمت ورودی و خروجی

ورودی :

- در خط اول به ترتیب از چپ به راست تعداد منابع موجود در سیستم برای R1, R2, R3 قرار می‌گیرد
- در خط بعدی تعداد وظیفه‌هایی که قرار است زمانبندی شوند وارد می‌شود.
- از خط سوم به بعد وظیفه‌ها به فرمت TaskName TaskType TaskDuration وارد می‌شوند، به عنوان مثال :

T1	Y	3
T2	X	6
T3	X	5
T4	Z	1

خروجی :

- بعد از هر واحد زمان، وضعیت دو صف موجود در سیستم، تعداد منابع موجود و وضعیت هر پردازنده (شامل وظیفه در حال اجرا بر روی آن) باید نمایش داده شود.
- در صورتی که امکان قرارگیری تسکی بر روی پردازنده وجود نداشته باشد، رو به روی آن Idle به معنی بیکار بودن قرار می‌گیرد.

- مثال:

```
R1: 0      R2: 2      R3: 0
priority queue: T2-T1-T3
waiting queue: T6
CPU1: T5
CPU2: T4
CPU3: Idle
CPU4: T7
```

توضیحات تکمیلی

- پروژه در گروه‌های ۲ نفره قابل انجام است. آپلود فایل توسط یکی از اعضای گروه کافی می‌باشد.
- استفاده از زبان‌های C, C++, python, java مجاز است (در صورت استفاده از C/C++ از makefile استفاده شود).
- مراحل پیاده‌سازی و نحوه اجرای برنامه خود را در فایل readme.md توضیح دهید.
- یک نمونه ورودی و خروجی از برنامه خود را در قالب دو فایل in.txt و out.txt ذخیره کنید.
- کد خود را به همراه فایل‌های خواسته شده در قالب یک فایل با فرمت `Scheduler_<student_names>_<student_ids>.zip` در ویو بارگذاری کنید.
- مهلت تحویل: جمعه ۱۷ دی، ۲۳:۵۹

موفق باشید