به نام خداوند رنگین کمان

درس شبیه سازی کامپیوتر دکتر صفائی

نیم‌سال اول ۱۴۰۲-۱۴۰۱

دانشکده مهندسی کامپیوتر

پروژه پایانی

مهلت ارسال:

به موارد زیر توجه کنید:

1- پروژه را می توانید در گروه های حداکثر ۲ نفره انجام دهید. انجام انفرادی پروژه نمره اضافه در پی نخواهد داشت.

2- گزارش و کد پروژه را در قالب یک فایل zip. با شماره دانشجویی و نام و نام‌خانوادگی خود، نام‌گذاری کرده و در سامانه CW بارگذاری کنید. در فایل گزارش (فرمت pdf.) نام تمام اعضای گروه ذکر شود.

3- در نوشتن گزارش به جواب آخر اکتفا نکنید، توضیحات کافی بنویسید.

4- تحویل پروژه به صورت مجازی خواهد بود و حضور تمامی اعضای گروه در جلسه تحویل الزامی است. در صورت غیبت در جلسه تحویل، نمره کسر خواهد شد.

5- در صورت انجام پروژه به صورت گروهی، تمامی اعضای گروه باید به مباحث تئوری تسلط کافی داشته باشند و به سوالات دستیاران آموزشی پاسخ دهند.

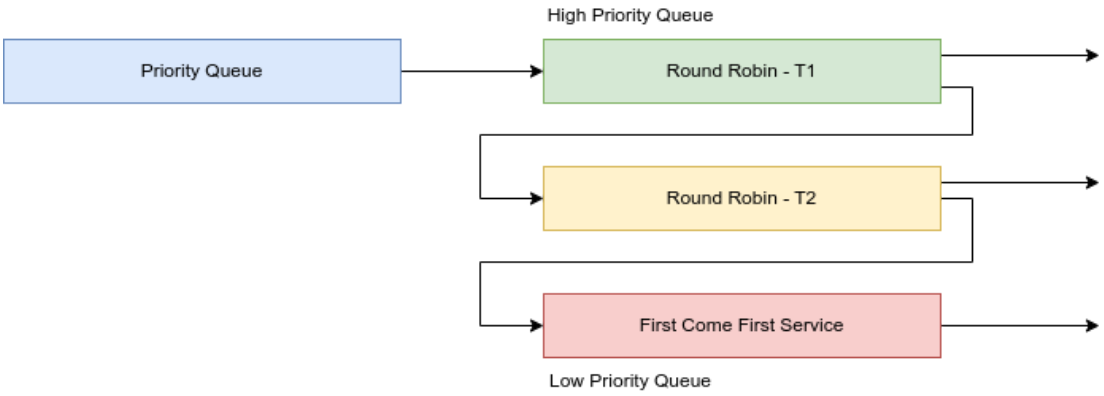
6- در صورت مشاهده هرگونه مشابهت نامتعارف، هر دو (یا چند) گروه کل نمره پروژه را از دست خواهند داد.

7- نمره نهایی از ۱۲۰ نمره است که ۱۰۰ نمره آن از بخش اجباری و ۲۰ نمره آن از بخش امتیازی است. این پروژه دو بخش امتیازی دارد که هر بخش ۱۰ نمره دارد.

در این پروژه شما باید یک زمان‌بند CPU را به صورت رخداد گسسته شبیه‌سازی کنید و خروجی‌های مورد نظر برای ارزیابی سیستم را بدست آورید.

توضیح سیستم

در این پروژه شما یک زمان‌بند CPU را شبیه‌سازی می‌کنید. پردازه‌ها در این سیستم همان تسک‌هایی هستند که در صف‌های متعدد جابه‌جا می‌شوند تا به CPU برسند (در اینجا Single-Core CPU). سیستم صف دو لایه است که در لایه اول یک Priority Queue داریم و در لایه دوم ۳ صف با سیاست‌های مختلف وجود دارد. در ابتدا پردازه‌ها وارد Priority Queue لایه اول می‌شوند و سپس بر اساس اولویتشان وارد صف اول لایه دوم می‌شوند. نحوه انتقال پردازه‌ها بین صف‌های این لایه در ادامه توضیح داده شده است.



جزئیات پیاده‌سازی

فرایند ایجاد تسک (لایه اول)

تسک‌های ورودی سیستم شما توسط متدی بنام JobCreator ساخته می‌شوند. این متد در یک فرایند پواسون با نرخ X تسک‌هایی تولید می‌کند که زمان سرویس آنها از توزیع نمایی با میانگین Y بدست می‌آید. همچنین اولیت هر تسک به صورت تصادفی با توجه به جدول زیر مشخص می‌شود. در نهایت تسک‌های تولید شده در Priority Queue لایه نخست قرار می‌گیرند.

اولیت تسک	Low	Normal	High
احتمال	0.7	0.2	0.1

فرایند انتقال به لایه دوم

با شروع فرایند شبیه‌سازی در بازه‌های ثابت زمانی مجموع تسک‌های موجود در در ۳ صف لایه دوم چک می شود. در صورتی که تعداد تسک‌ها کمتر از k باشد متدی به عنوان JobLoader فراخوانی شده و k تسک با بالا ترین اولیت و بالاترین زمان انتظار را به اولین صف لایه دوم یعنی RR-T1 منتقل میکند. در این صف اجرا تسک‌ها توسط CPU شروع می‌شود.

فرایند اجرای تسک

در فرایند شبیه‌سازی متدی به عنوان dispatcher تعریف میشود. این متد از بین ۳ صف لایه دوم بر اساس اولویت آنها یک صف را انتخاب کرده و بر اساس سیاست صف انتخابی یک تسک از آن را برداشته و در اختیار پردازنده قرار می‌دهد. در ادامه سیاست‌های این ۳ صف برای شما توضیح داده شده است.

- FCFS : همان صف FIFO است.
- Round-Robin-T1: این صف مانند FCFS عمل میکند با این تفاوت زمان اجرای هر تسک توسط پردازنده تا T1 واحد زمانی محدود می شود. یعنی اگر زمان سرویس یک تسک N واحد زمانی باشد تنها حداکثر T1 واحد زمانی در هر نوبت اجرا میشود.
- Round-Robin-T2: این صف مانند Round-Robin-T1 است و تنها تفاوت انها در محدودیت زمان اجرای تسک ها یا کوانتوم تایم آن است.

ممکن است زمان اجرای لازم برای تعدادی از تسک ها بیشتر از کوانتوم تایم صف های Round Robin باشد. در این صورت این تسک ها به لایه های پایینتر لایه دوم منتقل می شوند. یعنی اگر تسک در صف Round Robin-T1 باشد در صورت تمام نشدن وارد صف Round Robin-T2 میشود و اگر در این صف نیز زمان مورد نیاز بیشتر از کوانتوم تایم باید به صف لایه پایینتر یعنی FCFS منتقل می شود تا زمانی که اجرای آن شروع شود.

موارد امتیازی

برای بخش امتیازی می‌توانید موارد زیر را پیاده‌سازی کنید. در بخش اول کافیست فرایند انتخاب صف لایه ی دوم توسط dispatcher را بجای اولویت محوری به صورت تصادفی انجام دهید. در بخش دوم نیز کافیست برای پردازه‌های سیستم زمانی (timeout) را مشخص کنید که اگر پردازه بیشتر از آن زمان در حال انتظار بود، دیگر پردازش نشود.

اولویت صف‌های زمان‌بند

در حالت عادی صف‌های لایه دوم را بر اساس اولویتشان انتخاب می‌کنیم که این باعث می‌شود در صورت خالی نبودن صف‌های با اولویت بالا سایر تسک‌های صف‌های دیگر که اولویت کمتری دارند هیچ وقت انتخاب نشوند. این مسئله در صورتی که هیچ وقت صف‌ها خالی نشوند مشکلات جدی‌ای از جمله starvation و سرریز شدن صف را منجر میشود. برای جلوگیری از این مشکلات می‌توانیم در هر نوبت بطور تصادفی صف لایه دوم را انتخاب کنیم.

انتخاب صف لایه دوم	Round-Robin-T1	Round-Robin-T2	FCFS
احتمال	0.8	0.1	0.1

حداکثر زمان انتظار پردازه‌ها

راه دیگر برای جلوگیری از starvation، استفاده از timeout برای سیستم است. به این صورت که اگر یک پردازه زمان انتظارش بیشتر از timeout شود دیگر پردازشی روی آن صورت نمی‌گیرد و از سیستم بیرون انداخته می‌شود. هر پردازه، حداکثر مدت زمانی برای انتظار خواهد داشت و اگر بیشتر از آن زمان منتظر بماند، آن را از سیستم بیرون می‌اندازیم. این زمان از یک توزیع نمایی با میانگین Z پیروی می‌کند.

ورودی‌ها

این پارامترها به عنوان ورودی به شما داده می‌شوند:

- پارامترهای X, Y, Z (پارامتر Z امتیازی است)
- تعداد پردازه‌ها
- مدت زمان شبیه‌سازی (به ثانیه)

خروجی‌ها

یک توضیح مختصر از کد زده شده به همراه نحوه اجرای آن تهیه کنید. همچنین بعد از شبیه‌سازی این سیستم، به ازای هر دسته از ورودی‌های داده شده، این خروجی‌ها را در گزارش خود بیاورید:

- میانگین طول صف‌ها
- میانگین زمان صرف شده در صف‌ها (به طور کلی و به ازای هر پردازه)
- میزان بهره‌وری CPU (چه کسری از زمان مشغول بوده است؟)
- پیشنهاد خود را برای بهبود میانگین زمان صرف شده در صف‌ها را بنویسید (زمان کوانتوم دو صف اول لایه دوم را چگونه تغییر دهیم؟)
- درصد پردازه‌های منقضی شده (امتیازی)

نکات پیاده‌سازی

- در انتخاب زبان برنامه نویسی (مانند MATLAB, Java, Python و ..) برای پیاده سازی آزاد هستید.
- بهتر است ابتدا ساختار شی گرای مناسبی برای موجودیت‌های سیستم در نظر گرفته شود.
- در صورت استفاده از پایتون می توانید از کتابخانه های SimPy، Ciw و ... استفاده کنید.
- استفاده از کتابخانه های دیگر، به گونه ای که شبیه‌سازی را ساده تر کند، توصیه می شود.
- برای راحتی در پیاده‌سازی، زمان‌ها را گسسته‌سازی کنید، یعنی اگر مدت زمان یک سرویس به شکل عددی دارای اعشار محاسبه شد، آن را گرد کنید. همچنین همه‌ی واحدهای زمانی را ثانیه در نظر بگیرید.
- به‌عنوان چند مثال و آشنایی بیشتر با شبیه سازی در پایتون می توانید این لینک ها را مشاهده کنید (مسئله های مطرح شده در این منابع با مسئله‌ی ذکر شده تفاوت دارد، هدف اصلی آشنایی با معماری کد و نحوه پیاده سازی‌ها است). (لینک 1 و لینک 2 و لینک 3 و لینک 4)
- برای مثال، چند لینک زیر برای آشنایی با کتابخانه‌ی SimPy در پایتون به شما معرفی شده است و می توانید از آنها استفاده نمایید. (لینک 1 و لینک 2 و لینک 3)
- برای آشنایی با کتابخانه‌ی Ciw نیز میتوانید به این لینک مراجعه کنید.