

تمرین سری پنجم درس سیستم‌های عامل

فصل ۶ کتاب

موعد تحویل: ۱۳۹۴/۰۸/۲۲

- این تمرین را به صورت انفرادی انجام دهید.
- در صورت مشاهده ی هر گونه شباهت بین پاسخ‌ها نمره ی هر دو نفر ۰ در نظر گرفته می شود.
- هر گونه رونوشت از هر منبعی (همکلاسی‌ها ، اینترنت و ..) ممنوع می‌باشد و هر فرد بایستی فهم خود را از سوال مربوطه بنویسد.
- در صورت استفاده از کتاب یا منبعی از اینترنت حتماً نام و نشانی آن منبع را درج نمایید.
- پاسخ سوال‌های توضیحی را فقط به زبان فارسی بنویسید.
- کد مربوط به سوال ۶ و اسکرین شات‌های آن را در pdf جواب ضمیمه کنید.

۱. چرا سیاست shortest job first سیاستی بهینه است؟ (به بیان دیگر average waiting time در آن کینه است)
۲. پنج فرآیند زیر را در نظر بگیرید. به وسیله نمودار گانت، average turnaround time و average waiting time را برای سیاست‌های زیر محاسبه کنید (در شرایط یکسان، اولویت با فرآیندی است که آخر آمده باشد):

الف) FCFS with preemption

ب) Round Robin ($q = 3$)

ج) shortest job first with preemption

| | arrival time | burst time |
|----|--------------|------------|
| P1 | ۰ | ۱۰ |
| P2 | ۵ | ۱۵ |
| P3 | ۱۲ | ۱ |
| P4 | ۱۳ | ۱۵ |
| P5 | ۶۰ | ۷ |

د) یکی از الگوریتم‌هایی که در بعضی از مسائل می‌تواند کارایی بهتری نسبت به SJF داشته باشد، الگوریتم زمان‌بندی به کمک دانش از آینده است. به عبارتی پردازنده از لحظه ورود فرآیندها شروع به زمان‌بندی نمی‌کند بلکه به مدت زمان t_0 به صورت بیکار قرار می‌گیرد و سپس الگوریتم SJF را اجرا می‌کند. average waiting time فرآیندها را برای این حالت و به ازای $t_0 = 3$ محاسبه کنید.

۳. کدام یک از الگوریتم‌های زمان‌بندی زیر می‌تواند باعث قحطی شود؟ چرا؟

الف) FCFS

ب) SJF

ج) RR

د) Priority

۴. الگوریتم زمان‌بندی CFS را که در هسته لینوکس استفاده می‌شود را در حد چند خط معرفی کنید.

۵. چرا برای زمان‌بند مهم است که بداند برنامه مورد نظر IO-bound یا CPU-bound است؟

۶. در این تمرین هدف توضیح مفهوم process affinity است. در ابتدا یک برنامه بنویسید شامل یک حلقه نامتناهی باشد. این برنامه را اجرا کرده و وضعیت پردازنده‌ها را توسط یک برنامه مانند htop نظاره کنید. به صورت طبیعی می‌بایست در برهه‌های زمانی مختلف یکی از هسته‌های پردازنده‌های شما بار ۱۰۰٪ داشته باشد. در ادامه با استفاده از دستور sched_setaffinity() فرآیند خود را روی یک از هسته‌های پردازشی خود ثابت کرده و با استفاده از روشی که پیشتر از ذکر شد از کارکرد درست برنامه خود مطمئن شوید. توجه کنید که برنامه دوم شما در هر برهه‌ی زمانی تنها بار هسته محاسباتی که تنظیم شده است، را ۱۰۰٪ می‌کند.
(از مشاهدات خود در رابطه با بار پردازنده‌ها در قسمت الف و ب عکس (اسکرین شات) تهیه کنید).

۷. تفاوت زمان‌بندی‌های PCS و SCS را توضیح دهید.

۸. توضیح دهید چرا در سیستم‌های بی‌درنگ برای تاخیرهای dispatch و interrupt یک سقف در نظر می‌گیرند.

Trust The Computer. The Computer is your friend.
