به نام خدا

امتحان میانترم دوم سیستم های عامل **مصطفی فضلی شهری** – 9822803 1400/3/23

-1

بله در پردازنده های نسل جدید راه حل پیترسون به مشکل برخورد کرده و نمیتوان از آن استفاده کرد، اما در پردازنده های قدیمی این راه حل می تواند بدون مشکل انجام شود اینگونه که در درس اشاره شد می توان بدین صورت بیان کرد که به ترتیب اسلاید ها و توضیح استاد:

- پرچم flags تعیین می کند که آیا یک فرآیند میخواهد وارد ناحیه بحرانی شود یا خیر.
- اگر flags[0] == true باشد، P0 می‌خواهد وارد ناحیه بحرانی شود.
- اگر flags[1] == true باشد، P1 می‌خواهد وارد ناحیه بحرانی شود.
- مقدار اولیه برای هر همه flags ها false است.
- با استفاده از راه حل پیترسون در ابتدا هر فرآیند با تبدیل flags خود به true اعلام میکند که می خواهد وارد ناحیه بحرانی شود.
- اما نوبت را به فرآیند دیگر میدهد و مقدار turn را برابر با شماره فرآیند دیگر میکند و تعارف میکند.
- در این هنگام بررسی میکند که اگر فرایند دیگر از ناحیه بحرانی خارج شده یا وارد ناحیه بحرانی نشده باشد، خودش وارد ناحیه بحرانی بشود.
- در این هنگام کد ها از فرآیند های P0 و P1 به صورت خط به خط اجرا می شوند.

در پردازنده های نسل جدید که در یک کلاک ممکن است چندین دستور انجام شود یا ترتیب انجام دستورات جابجا شود، از طرفی ممکن است کامپایلر در روند اجرای دستورات تغییر ایجاد کند، نمیتوان از راه حل پیترسون استفاده کرد و باید از راه حل های دیگر استفاده کرد.

-2

خیر این راه حل خاصیت انتظار محدود ندارد زیرا :

هیچ تضمینی نداریم که وقتی یک فرآیند وارد ناحیه بحرانی بشود و از آن خارج بشود، دوباره خود آن فرآیند وارد ناحیه بحرانی نشود، بدین صورت ممکن است بقیه فرآیند ها نتوانند به نوبت خود برسند.

-3

الف) الگوریتم موازی هنگامی کارآمد است که زیرساخت های لازم برای آن را داشته باشیم؛ مثلا هنگامی که چندین پردازنده یا پردازنده چند هسته ای داشته باشیم که بتوانند بخش های موازی الگوریتم همزمان اجرا کنند و زمان اجرای برنامه را کاهش دهند.

(البته نیاز است تا همه زیرساخت های لازم را برای این الگوریتم داشته باشیم، در غیر این صورت در سخت افزاری که این امکان را ندارد یا در آن دچار ضعف است فشار زیاد یا به اصطلاح حالت گلوگاه رخ می دهد.)

ب) بله، در صورتی که پردازنده مورد استفاده تک هسته ای باشد، الگوریتم سری سریع تر از الگوریتم موازی انجام می شوند؛ زیرا بخش های مربوط به سوئیچ بین فرآیند ها و پردازنده ها انجام نمی شود و از به جای تایم اتلاف وقت سوئیچ بین فرآیند ها، فرآیند ها به طور کامل انجام می شوند.

-4

Parent = A

Child = B

الف) هنگامی که فرآیند والد بسته می شود ولی فرآیند B در حال اجرا باشد، حالت Orphan رخ می دهد، اینجا نیز هنگامی که فرآیند B که فرزند می باشد، در حال اجرا است فرآیند A که فرآیند والد است بسته می شود و این حالت رخ می دهد.

در این حالت یک فرآیند دیگر که معمولا اولین فرآیند (در ریشه که مثلا systemd است) جایگزین والد حذف شده قبلی آن می شود.

ب) هنگامی که فرآیند والد در حال اجرا باشد و فرآیند فرزند بسته شود:

منابع مربوط به فرآیند فرزند (مثل حافظه اختصاص داده شده و پورت های شبکه باز شده و...) آزاد می شوند و مقدار exit code که داده مهم و لازم برای فرآیند والد است که صفر برای اجرای بدون مشکل و موفقت آمیز است در PCB آن نوشته می شود.

تا هنگامی که فرآیند والد، فراخوانی سیستمی wait را اجرا نکند، مقدار exit code را دریافت نکرده و PCB آن حذف نمی شود، در این هنگام این فرآیند در حالت زامبی (Zombie) قرار می گیرد.

هنگامی که والد فراخوانی سیستمی wait را اجرا می کند، مقدار exit code را دریافت می کند و PCB فرآیند حذف می شود و فرآیند از حالت زامبی خارج می شود.