

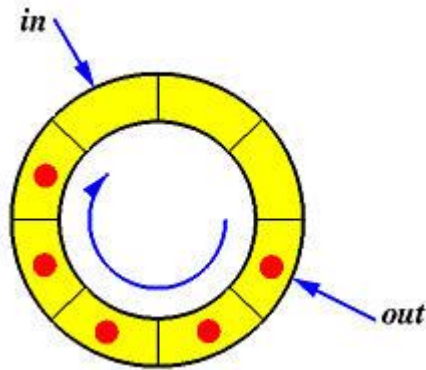
حل تمرین پنجم درس سیستم‌های عامل

دکتر زرندی

پاییز ۹۹

۱- در روش تبادل پیام از بافر با ظرفیت ۱۰ پیام استفاده کرده‌ایم. فرستنده هر ثانیه یک پیام آماده ارسال دارد و گیرنده برای پردازش یک پیام به ۱.۵ ثانیه زمان نیاز دارد. چقدر طول می‌کشد تا فرستنده تمام پیام‌های خود را در بافر قرار دهد؟ پاسخ خود را با توجه به مساله تولیدکننده-مصرف‌کننده شرح دهید.

در بافر چرخه‌ای با ظرفیت n می‌توان حداکثر $n-1$ پیام قرار داد.



در ثانیه ۱، تولیدکننده یک پیام در بافر می‌گذارد و مصرف‌کننده آن را برمی‌دارد تا پردازش بر روی آن انجام شود. از اینجا به بعد هر ۳ ثانیه ۳ پیام در بافر گذاشته می‌شود و ۲ پیام برداشته می‌شود. پس هر ۳ ثانیه یک پیام به بافر اضافه می‌شود.

$$\frac{t-1}{3} = 9 \rightarrow t = 28$$

در ثانیه ۲۸ بافر پر می‌شود. از اینجا به بعد تولیدکننده باید برای مصرف‌کننده صبر کند. بنابراین هر ۱.۵ ثانیه یک پیام در بافر می‌گذارد.

$$28s + (100 - 28) \times 1.5s = 136s$$

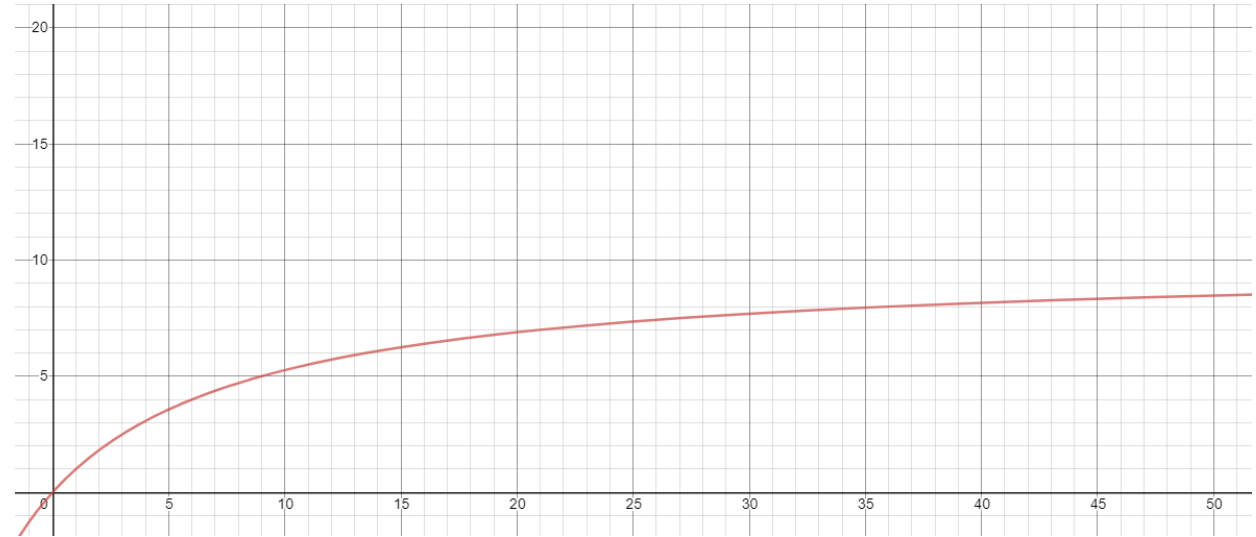
۲- یک الگوریتم داریم که ۹۰٪ قابلیت موازی سازی دارد. برای اینکه به ۸۰٪ از حداکثر میزان تسریع تئوری دست پیدا کنیم، باید حداقل از چه تعداد ریسمان در اجرای این الگوریتم استفاده کرد؟

Amdahl's Law:

$$S_{\text{latency}}(s) = \frac{1}{(1 - p) + \frac{p}{s}}$$

Maximum Theoretical Speedup:

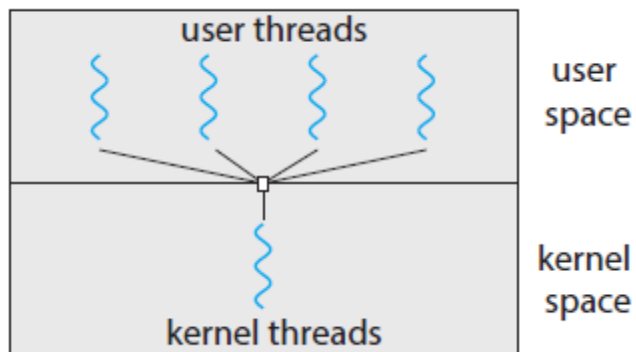
$$\lim_{s \rightarrow \infty} S_{\text{latency}}(s) = \frac{1}{1 - p}$$



$$\frac{1}{(1 - 0.9) + \frac{0.9}{s}} = 0.8 \times \frac{1}{1 - 0.9}$$

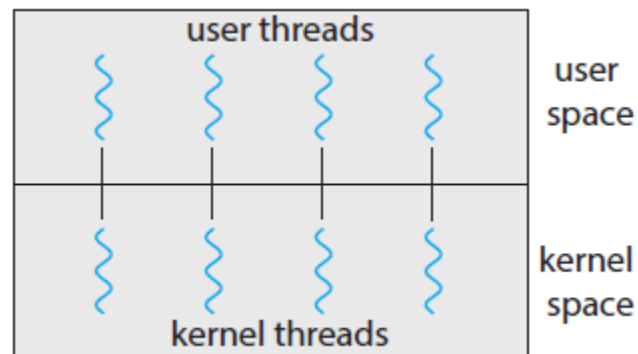
$$s = 36$$

۳- مدل‌های مختلفی برای ارتباط میان ریسمان‌های هسته و ریسمان‌های کاربر وجود دارد:
الف) در چه حالتی مدیریت ریسمان‌ها بهینه‌تر می‌باشد؟ در چه روشی محدودیتی برای تعداد ریسمان‌های هسته وجود ندارد؟ آیا این روش‌ها در سیستم‌عامل‌های امروزی کاربرد دارند؟ چرا؟



در مدل چند به یک مدیریت ریسمان‌ها بهینه می‌باشد.
زیرا توسط کتابخانه ریسمان موجود در فضای کاربر انجام می‌شود.

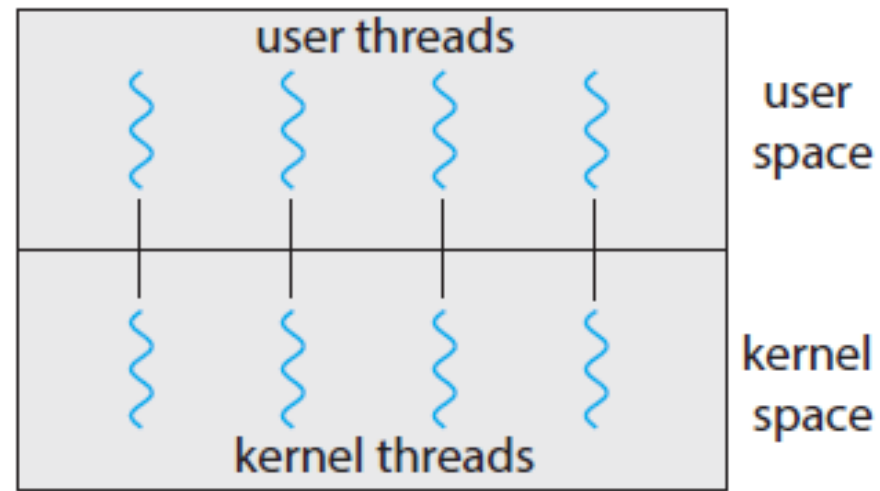
در واقعیت تعداد ریسمان‌های هسته نمیتواند از حدی بیشتر شود اما در مدل یک به یک طبق تعریف به ازای هر ریسمان کاربر یک ریسمان هسته ایجاد می‌شود. اگر کاربر تعداد زیادی ریسمان ایجاد کند، می‌تواند باعث کند شدن سیستم شود.



امروزه مدل چند به یک متداول نیست زیرا قادر به بهره‌وری از پردازنده‌های چند هسته‌ای نیست. در عوض از مدل یک به یک در بیشتر سیستم‌عامل‌های امروزی استفاده می‌شود، زیرا اجازه اجرای تعداد زیادی ریسمان را می‌دهد. همچنین با افزایش تعداد هسته‌های پردازشی دیگر محدود کردن تعداد ریسمان‌های هسته اهمیت چندانی ندارد.

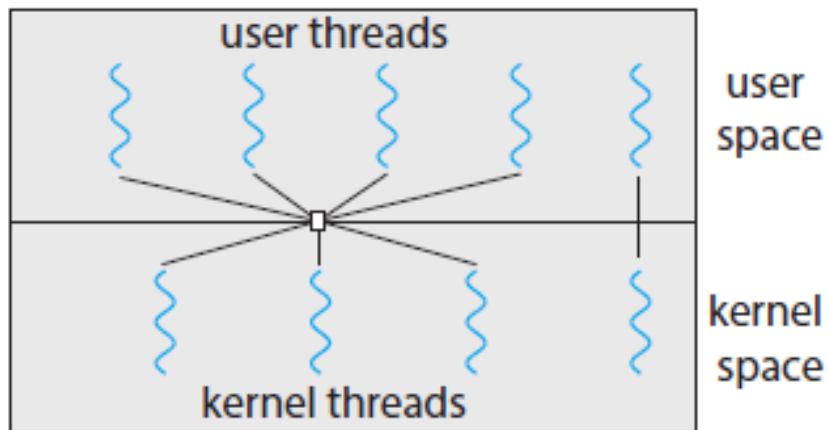
ب) در مدل یک به یک چه احتیاطی باید رعایت شود؟ چرا؟

در مدل یک به یک، کاربر در مورد تعداد ریسمان‌هایی که ایجاد می‌کند احتیاط کند.
زیرا به ازای هر ریسمان کاربر یک ریسمان هسته ایجاد می‌شود.
تعداد زیاد ریسمان‌های هسته می‌تواند موجب کند شدن سیستم شود.



ج) مزیت‌های مدل چند به چند چیست؟ چرا در سیستم‌های عامل امروزی از این روش استفاده نمی‌شود؟

در مدل چند به چند، بر خلاف مدل چند به یک، می‌توان از چندین ریسمان هسته استفاده کرد که باعث هم‌رندی بیشتر و بهره‌وری از هسته‌های پردازشی می‌شود.



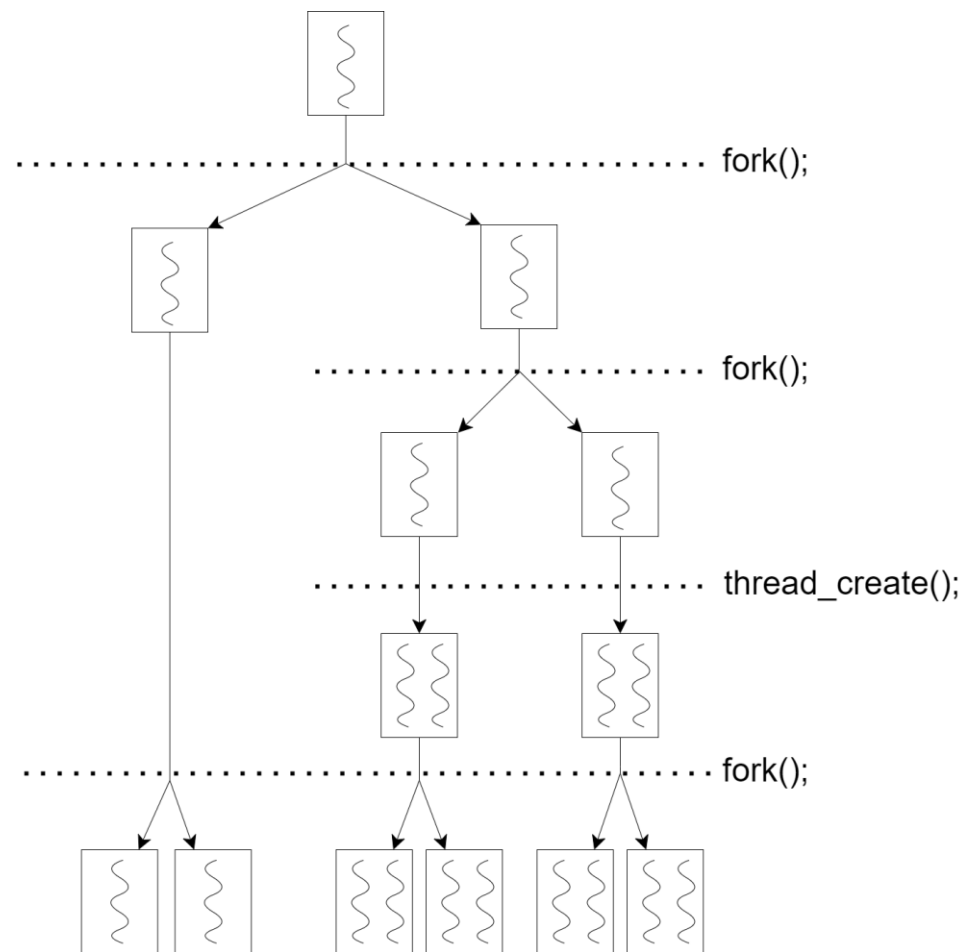
همچنین بر خلاف مدل یک به یک، با افزایش تعداد ریسمان‌های کاربر سیستم دچار مشکل نشده و نیازی نیست که کاربر نگران این موضوع باشد.

پیاده‌سازی مدل چند به چند از بقیه مدل‌ها پیچیده‌تر می‌باشد. علاوه بر آن با افزایش تعداد هسته‌های پردازشی در سیستم‌های امروزی، دیگر نیازی به محدود کردن تعداد ریسمان‌های هسته نیست. به همین دلیل اکثر سیستم‌عامل‌های امروزی از مدل یک به یک استفاده می‌کنند.

۴- در قطعه کد زیر چه تعداد فرآیند و ریسمان متمایز ساخته می‌شود؟

```
pid_t pid;  
  
pid = fork();  
if (pid == 0) { /* child process */  
    fork();  
    thread_create( . . . );  
}  
fork();
```

6 processes & 10 threads



۵- در یک مرورگر کاربر دکمه توقف را فشار داده است و همه‌ی ریسمان‌های صفحه مورد نظر باید لغو شوند.
الف) در صورت استفاده از روش کنسل کردن ناهمگام چه مشکلی ممکن است رخ دهد؟ چرا؟

در صورت استفاده از کنسل کردن ناهمگام، ممکن است بعضی از منابعی که مورد استفاده ریسمان‌های مربوط به بارگذاری صفحه هستند، آزاد نشده تا در اختیار فرآیندهای بعدی قرار بگیرند. زیرا این ریسمان‌ها فرصت آزادسازی منابع را نخواهند داشت. در صورت تکرار این روند، مقدار زیادی از منابع سیستم هدر خواهد رفت و از کنترل سیستم‌عامل خارج می‌شود.

ب) فرض کنید که از روش کنسل کردن موخر استفاده کرده‌ایم. توضیح دهید چگونه می‌توان مطمئن شد که همه ریسمان‌ها در زمان نسبتاً کوتاهی لغو می‌شوند؟

برای اطمینان از لغو شدن به موقع ریسمان‌ها باید آن‌ها را به گونه‌ای طراحی کرد که پس از مدت کوتاهی از صدا کردن تابع لغو، ریسمان‌ها به نقطه کنسلی برسند. در این نقطه منابع مصرف شده توسط ریسمان آزاد شده و سپس ریسمان لغو می‌شود. همچنین می‌توان به صورت ترکیبی از روش ناهمگام و موخر استفاده نمود.