

موسسه بابان

انتشارات بابان و انتشارات راهیان ارشد

درس و کنکور ارشد

سیستم عامل

(مدیریت نخ)

ویژه‌ی داوطلبان کنکور کارشناسی ارشد مهندسی کامپیوتر و IT

بر اساس کتب مرجع

آبراهام سیلبرشاتز، ویلیام استالینگز و اندرو اس تنن‌بام

ارسطو خلیلی‌فر

کلیه‌ی حقوق مادی و معنوی این اثر در سازمان اسناد و کتابخانه‌ی ملی ایران به ثبت رسیده است.

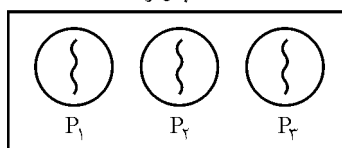
مدیریت نخ

۳

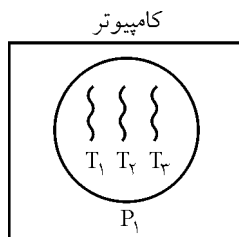
نخ (Thread)

در سیستم عامل‌های قدیمی‌تر، به ازای هر فرآیند یک رشته نخ یا رشته اجرایی و به تبع یک شمارنده برنامه (PC) وجود داشت اما در سیستم عامل‌های امروزی به ازای هر فرآیند می‌توان چند نخ یا رشته اجرایی داشت. شکل زیر سه فرآیند معمولی را نشان می‌دهد که هر یک برای خودشان یک رشته اجرایی و یک حافظه مختص به خود را دارند.

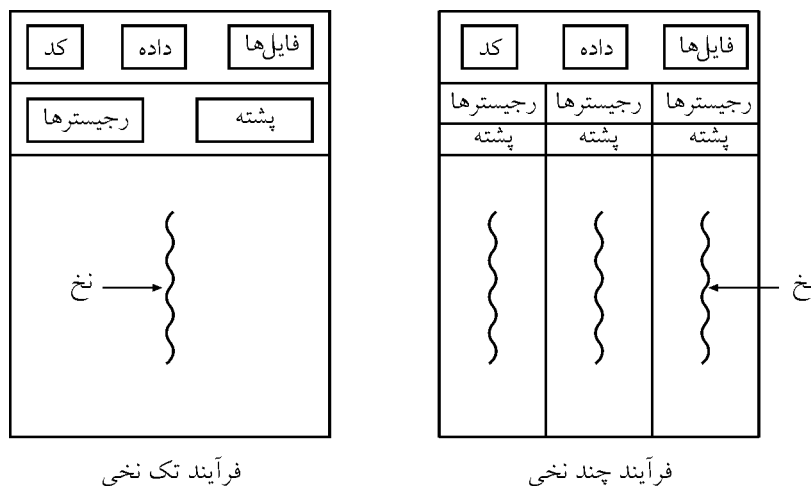
کامپیوتر



ولی در شکل زیر یک فرآیند، سه رشته اجرایی دارد که هر یک رجیستر، پشته و شمارنده برنامه (PC) مجزای خود را دارند و مانند فرآیندها می‌توانند همروند (در سیستم‌های تک پردازنده‌های) و موازی (در سیستم‌های چند پردازنده‌ای) اجرا شوند.



نکته: نخ‌های هم‌تا که در یک فرآیند قرار دارند، و از کد، داده و منابع مشترک استفاده می‌کنند اما هر نخ، شمارنده برنامه، مجموعه رجیستر و فضای پشته جداگانه‌ای در اختیار دارد. در واقع هر نخ، TCB مجزایی دارد.



نکته: از آن جا که نخ‌های هم‌تا در یک فرآیند قرار داشته و اشتراکات زیادی با هم دارند، عمل تعویض متن بین آنها به راحتی و با هزینه‌ی کمتری صورت می‌گیرد، در واقع TCB مربوط به نخ‌ها، محتوی کمتری نسبت به PCB فرآیندها دارد، مثلاً لیست فایل‌های باز مربوط به فرآیندها است، بنابراین این لیست به هنگام تعویض متن فرآیندها باید داخل PCB مربوط به فرآیند ذخیره گردد، در حالی که به هنگام تعویض متن بین نخ‌ها نیازی به ذخیره‌سازی لیست فایل‌های باز مربوط به یک فرآیند در TCB یک نخ نیست. بنابراین تعویض متن بین نخ‌ها به نسبت به فرآیندها ارزان‌تر است.

نکته: گاهی از نخ به عنوان Light Weight Process (فرآیند سبک وزن) نیز یاد می‌کنند و به کل یک فرآیند، Heavy Weight Process (فرآیند سنگین وزن) گویند.

نکته: نخ‌ها هم همانند فرآیندها می‌توانند حالت‌های مختلفی را تجربه کنند مانند آماده، در حال اجرا یا منتظر. در واقع پردازنده می‌تواند بین نخ‌ها به اشتراک گذاشته شود.

چند نخی در زبان C#

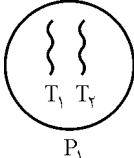
C# پیاده سازی مفهوم چند نخی را پشتیبانی می کند. در زبان C#، هر برنامه به طور پیش فرض از یک نخ تشکیل شده است. و در صورت ایجاد نخ های دیگر، مفهوم چند نخی پیاده سازی می گردد. توجه: نخ اول به صورت پیش فرض وجود دارد و برنامه با نخ اول شروع به اجرا می کند.

مثال: در قطعه کد زیر نخ T₁ به طور پیش فرض وجود دارد و نخ T₂ ایجاد می گردد.

```
static void main ()
{
    Thread T2=new Thread (Go) ;
    T2.Start()
    Go ( );
}
static void Go ( )
{
    for (int i = 0 ; i<5 ; i++)
        console.write("*");
}
```

↓
در این بخش نخ T₂ ایجاد می گردد و فعالیت Go داخل این نخ قرار می گیرد.

نخ T₂ اجرا می گردد. فعالیت Go داخل نخ T₁ قرار داده شده است. می شود فعالیت دیگری تعریف شود و در این بخش فراخوانی شود.



توجه: فعالیت Go داخل نخ T₁ قرار دارد، اما فعالیت Go در نخ T₂ هم قرار داده شده است. در دو نخ T₁ و T₂ که فعالیت Go داخل آن قرار دارد، متغیر محلی i در داخل پشته مربوط به هر نخ ایجاد می گردد.

بنابراین خروجی این برنامه به صورت زیر خواهد بود:

چاپ ۱۰ عدد ستاره به دلیل اجرای همروند (سیستم تک پردازنده ای) یا موازی (سیستم چند پردازنده ای) دو نخ T₁ و T₂ است!

مثال: در قطعه کد زیر نخ T_1 به طور پیش فرض وجود دارد و نخ T_2 ایجاد می‌گردد.

```
static void main()
{
    Thread T2=new Thread (func) ;
    T2.start( ) ;
    console.write("X");
}
static void func( )
{
    console.write("Y");
}
```

↓
در این بخش نخ T_2 ایجاد می‌گردد و فعالیت $func$ داخل این نخ قرار می‌گیرد.

نخ T_2 اجرا می‌گردد. فعالیت چاپ "X" داخل نخ T_1 قرار داده شده است.

بنابراین خروجی این برنامه به صورت زیر خواهد بود:

XY

مزایای فرآیندهای چند نخ

۱ - ساختار بسیاری از برنامه‌های کاربری ذاتاً از بخش‌های کاملاً مستقل تشکیل می‌شوند که جدا نکردن آن‌ها باعث پیچیدگی بالا و کاهش خوانایی در برنامه می‌گردد. مهندسی نرم‌افزار نیز بر ساخت برنامه‌های کاربردی توسط پیمان‌های مستقل نیز تأکید دارد. برای مثال، خطاست اگر بیندیشید که برنامه شبیه سازی ۱۱ بازیکن یک تیم فوتبال در یک نخ قرار گیرد. به عنوان مثالی دیگر یک برنامه‌ی واژه پرداز می‌تواند از نخ‌های مستقلی مانند کنترل املا و گرامر، صفحه آرایی، مدیریت ورودی‌های کاربر و غیره تشکیل شده باشد.

۲ - در فرآیند تک نخ، هرگاه فراخوان سیستمی مسدود کننده‌ای اجرا شود، کل فرآیند مسدود می‌گردد. در حالی که در فرآیندهای چند نخ در صورتی که سیستم عامل زمان‌بندی چند نخ را پشتیبانی کند، فقط نخ که فراخوان سیستمی مسدود کننده دارد، مسدود می‌گردد و مابقی نخ‌های یک فرآیند می‌توانند پس از در اختیار گرفتن پردازنده، اجرا گردند. فرآیند تک نخ مانند قانونی می‌باشد که اگر یک نفر در خانواده خطا کند، همه خانواده محکوم می‌گردند و فرآیند چند نخ مانند قانونی می‌باشد که اگر یک نفر در خانواده خطا کند، فقط همان یک نفر محکوم می‌گردد و بقیه خانواده می‌توانند به زندگی طبیعی خود ادامه دهند.

۳ - ایجاد همروندی (در سیستم تک پردازنده‌ای) و توازی (در سیستم چند پردازنده‌ای) در نخ‌های یک فرآیند و فرآیندهای دیگر.

مثال: کاربرد چند نخی در فرآیند سمت سرویس دهنده.

در این مدل، فرآیند سمت سرویس دهنده از چندین نخ جهت پاسخ به درخواست‌های کاربر یعنی سرویس گیرنده تشکیل شده است. پاسخ هر کاربر می‌تواند توسط یک نخ از سمت سرویس دهنده داده شود. چنانچه نخ‌ها در فرآیند سرویس دهنده جهت تبادل داده از روی دیسک به سمت سرویس گیرنده مسدود گردد، نخ‌های دیگر فرآیند سرویس دهنده می‌توانند به درخواست‌های دیگر، پاسخ دهند. زیرا کارکرد آن‌ها وابسته به نخ مسدود شده نیست.

توجه: شاید بگویید به جای قرار دادن کارهای مختلف یک سرویس دهنده در داخل نخ‌های یک فرآیند، می‌شد هر یک از کارها را در داخل یک فرآیند قرار داد و چند فرآیندی را در مقابل چند نخی ابداع کرد. اما به دلایل زیر استفاده از چند نخ معقولانه‌تر به نظر می‌رسد:

- هزینه زمانی ایجاد (بارگذاری TCB) و پایان دادن (ذخیره‌سازی TCB) به یک نخ در یک فرآیند به مراتب کمتر از ایجاد (بارگذاری PCB) و پایان دادن (ذخیره‌سازی PCB) یک فرآیند است. نخ‌های داخل یک فرآیند، از برخی منابع به صورت مشترک استفاده می‌کنند، در حالی که فرآیندها، منابع مختص به خود را در اختیار می‌گیرند.

- نخ‌های هم‌تا در یک فرآیند، اشتراکات زیادی با هم دارند، بنابراین عمل تعویض متن بین آنها با هزینه کمتری انجام می‌گردد. در حالی که تعویض متن بین فرآیندها به دلیل عدم اشتراکات با هزینه بیشتری انجام می‌گردد.

توجه: اشتراکات نخ‌های داخل یک فرآیند شامل سگمنت داده (داده سراسری)، فضای آدرس، فایل‌های باز و اختلاف نخ‌های داخل یک فرآیند شامل شمارنده برنامه (PC)، رجیسترها و پشته می‌باشد.

زمان بندی نخ‌ها

مدیریت و زمان بندی نخ‌ها به سه روش زیر انجام می‌گردد:

۱- روش سطح کاربر یا مدل چند به یک (many to one)

در این روش فقط زمان بند پردازنده و زمان بند چند نخی در سطح کاربر وجود دارد و زمان بند چند نخی در سطح هسته در این روش مورد استفاده قرار نمی‌گیرد. در واقع هسته سیستم عامل فقط فرآیندها را می‌شناسد و هیچ اطلاعاتی از نخ‌ها ندارد. در واقع اولویت بندی نخ‌ها، مدیریت نخ‌ها و زمان بند چند نخی در سطح کاربر و توسط یک بسته نرم‌افزاری انجام می‌گردد. بدین معنی که نخ‌ها را برنامه نویسنده مشخص می‌کند و مدیریت آن‌ها را نیز بر عهده می‌گیرد. بنابراین زمان بند پردازنده، بر اساس الگوریتم مشخصی مثلاً نوبت چرخشی پردازنده را در اختیار یکی از فرآیندهای آماده قرار می‌دهد. سپس زمان بند چند نخی در سطح کاربر، متناسب با کاربردی که در آن فرآیند به کار گرفته می‌شود، الگوریتم

زمان بندی را انتخاب کرده و تصمیم می‌گیرد که پردازنده در اختیار کدام یک از نخ‌های آماده در فرآیند مورد نظر قرار گیرد و تا زمانی که پردازنده در تملک فرآیند باشد و یا تا قبل از پایان برش زمانی مربوط به فرآیند، نخ‌های یک فرآیند از پردازنده بهره‌مند می‌شوند و به محض مسدود شدن یک نخ، و یا پایان برش زمانی فرآیند، یا اتمام فرآیند، پردازنده به فرآیند بعدی تعلق می‌گیرد.

توجه: در این روش نخ ماهیت منطقی دارد و از دید کاربر فقط وجود دارد، در واقع از نظر سیستم عامل ماهیت فیزیکی ندارد، بنابراین نخ کاربر در این روش همانند یک تابع در فرآیند می‌باشد که از رجیستر و پشته مختص به خود نیز بهره‌مند نمی‌باشد.

توجه: مدل غیر کامپیوتری این روش نیز وجود دارد، مانند حالتی که درآمدهای دولت حاصل از منابع کشور، بین پدران خانواده‌ها تقسیم گردد و این پدران خانواده‌ها باشند که تصمیم بگیرند به هر عضو خانواده چه مقدار نقدینگی تعلق بگیرد. در این روش فقط پدران شماره حساب مختص به خود را دارند. اما اگر یکی از اعضای خانواده خطایی انجام دهد و محکوم گردد، آنگاه تمام اعضاء خانواده، برای مدتی از خدمات دولت محروم می‌گردند، زیرا در این مدل، دولت فقط پدران خانواده‌ها را می‌شناسد و از اعضا خانواده اطلاعی ندارد. بنابراین حساب پدر خانواده برای مدتی مسدود می‌گردد. **توجه:** عمل تعویض متن مابین نخ‌های یک فرآیند کاربر، کاملاً در سطح کاربر و با سربرار بسیار ناچیز (در حد فراخوانی نخ بعدی) و بدون تغییر حالت پردازنده به مُد هسته پردازنده، توسط زمان‌بند چند نخی در سطح کاربر (برنامه‌های کاربر) انجام می‌گردد. اما عمل تعویض متن ما بین فرآیندها (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در سطح هسته سیستم عامل و در مُد هسته پردازنده، توسط زمان‌بند پردازنده برای انتخاب یک فرآیند جدید براساس یک الگوریتم خاص انجام می‌گردد.

توجه: به دلیل آنکه نخ‌های یک فرآیند کاربر، تماماً در فضای کاربر مدیریت می‌شوند، اگر نخ موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدود کننده را اجرا نماید، هسته سیستم عامل نه تنها آن نخ، بلکه کل فرآیند کاربر را که شامل تمام نخ‌های دیگر می‌باشد، مسدود می‌کند، زیرا هسته سیستم عامل خبری از نخ‌های داخل فرآیند کاربر ندارد. در واقع در این روش هسته سیستم عامل نخ‌ها را همانند توابع داخل یک فرآیند می‌بیند، یعنی هسته سیستم عامل، یک فرآیند چند نخی سطح کاربر را، مانند یک فرآیند تک نخی اما دارای چند تابع مختلف می‌بیند!

توجه: این راه کار، منجر به عدم امکان هم روندی (در سیستم‌های تک پردازنده‌ای) نخ‌های داخل یک فرآیند کاربر در حالت انسداد یک نخ داخل یک فرآیند کاربر می‌گردد.

البته اگر نخ‌های داخل یک فرآیند کاربر مسدود نگردد، امکان هم روندی میان نخ‌های داخل فرآیند کاربر برقرار است. مانند یک تیم فوتبال ۱۱ نفره که اگر بازیکنی مرتکب خطا گردد، از آن جا که داور فقط نام تیم را می‌شناسد و نه تک تک بازیکنان تیم را، آنگاه کل تیم را جریمه، اخراج و مسدود می‌کند. اما اگر هیچ یک از بازیکنان تیم مرتکب خطایی نگردند، واضح است که هم‌روندی برقرار است.

توجه: فرض کنید یک کیک داریم که آن را به چهار قسمت مساوی تقسیم کرده‌ایم، همچنین فرض کنید خوردن هر بخش کیک یک ساعت زمان بخواهد، اگر یک نفر بخواهد تمام این کیک را بخورد، پس ۴ ساعت طول می‌کشد و اگر ۴ نفر بخواهند تمام این کیک را بخورند و به هر نفر یک بخش کیک داده شود، آن‌گاه خوردن تمام کیک به طور موازی ۱ ساعت طول خواهد کشید. در روش سطح کاربر، یک فرآیند چند نخ‌ی کاربر نمی‌تواند از امتیازات چند پردازنده‌ای بهره‌بردار، زیرا در روش سطح کاربر، هسته سیستم عامل در هر لحظه فقط یک پردازنده را در اختیار نخ‌های یک فرآیند کاربر قرار می‌دهد. حتی اگر چند پردازنده موجود باشد. یعنی در این روش خوردن کیک بخش بندی شده به صورت چند نفری امکان‌پذیر نیست. بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند کاربر وجود ندارد.

توجه: نرم‌افزارهای Mach C-threads و POSIX P-threads به عنوان یک بسته نرم‌افزاری، می‌توانند جهت مدیریت نخ‌ها و زمان‌بند چند نخ‌ی در سطح کاربر مورد استفاده قرار گیرند. توجه: در این روش تخصیص منابع و زمان‌بندی پردازنده، بر روی فرآیندها انجام می‌شود. همچنین زمان‌بندی نخ‌های سطح کاربر، بر عهده زمان‌بند چند نخ‌ی سطح کاربر خواهد بود. توجه: سیستم عامل سولاریس، مدل چند به یک را پیاده‌سازی می‌کند.

۲- روش سطح هسته یا مدل یک به یک (one to one)

در این روش فقط زمان‌بند پردازنده و زمان‌بند چند نخ‌ی در سطح هسته وجود دارد و زمان‌بند چند نخ‌ی در سطح کاربر در این روش مورد استفاده قرار نمی‌گیرد. در واقع هسته سیستم عامل نه تنها فرآیندها را می‌شناسد بلکه از وجود نخ‌های داخل یک فرآیند نیز در صورت وجود نخ، آگاه است. در واقع مدیریت نخ‌ها و زمان‌بندی چند نخ‌ی (نخ‌های فرآیندهای کاربر و نخ‌های فرآیندهای سیستم عامل) در سطح هسته، توسط هسته سیستم عامل انجام می‌گردد و کاربر هیچ دیدی از این کار ندارد. انگار که اجتماع تمام نخ‌های تمام فرآیندهای سیستم عامل و کاربر را در نظر بگیرید، حال بر روی تک تک نخ‌ها بر اساس یک الگوریتم خاص حرکت کنید. بنابراین زمان‌بند چند نخ‌ی در سطح هسته، بر اساس الگوریتم مشخصی مثلاً نوبت چرخشی پردازنده را در اختیار یکی از نخ‌های آماده قرار می‌دهد. توجه کنید که در این روش پردازنده دیگر در تملک فرآیند نیست بلکه در تملک نخ‌ها است. در واقع تا زمانی که پردازنده در تملک یک نخ باشد و تا قبل از مسدود شدن نخ، و یا تمام شدن نخ و یا پایان برش زمانی مربوط به نخ، نخ می‌تواند از پردازنده بهره‌بردار و به محض مسدود شدن یک نخ، یا پایان برش زمانی مربوط به نخ و یا اتمام نخ، پردازنده می‌تواند به نخ بعدی که ممکن است، نخ بعدی، نخ‌ی هم‌خانواده با نخ قبلی در یک فرآیند باشد، یا نخ‌ی در یک فرآیند دیگر باشد، تعلق بگیرد. در واقع زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل تعیین می‌کند که نخ بعدی که باید شروع به کار کند متعلق به همان فرآیند

باشد یا از یک فرآیند دیگر انتخاب شود.

توجه: برای انجام زمان‌بندی چند نخ، هسته سیستم عامل باید علاوه بر جدول فرآیندها، یک جدول نخ (شبه جدول فرآیند) داشته باشد که اطلاعات تمامی نخ‌های موجود در سیستم را نگهداری کند. مجدداً تأکید می‌کنیم که در این حالت هر نخ TCB خاص خود را دارد. به عبارت دیگر هر نخ رجیسترها و پشته مختص به خود را دارد.

توجه: در این روش نخ ماهیت فیزیکی دارد، و از دید کاربر و سیستم عامل وجود دارد، بنابراین در این روش، هر نخ، رجیستر و پشته مختص به خود را دارد، به عبارت دیگر هر نخ TCB مختص به خود را دارد.

توجه: عملیات مرتبط با نخ‌های سطح کاربر، مانند ایجاد (بارگذاری TCB مختص به نخ) و پایان دادن (ذخیره‌سازی TCB مختص به نخ) بر عهده هسته سیستم عامل است و توسط زمان‌بند چند نخی در سطح هسته سیستم عامل انجام می‌گردد.

توجه: مدل غیر کامپیوتری این روش نیز وجود دارد، مانند حالتی که درآمدهای دولت حاصل از منابع کشور، بدون اعمال هیچ‌گونه اولویت‌بندی بین تک تک اعضای یک کشور تقسیم گردد (حال این اعضا شهروند عام باشند یا خاص) در این روش هر یک از اعضای کشور شماره حساب مختص به خود را دارند. اما اگر یکی از افراد کشور خطایی انجام دهد و محکوم گردد، آنگاه فقط همان فرد برای مدتی از خدمات دولت محروم می‌گردد و دولت به بقیه اعضای خانواده آن فرد همچنان خدمات ارائه می‌دهد. زیرا دولت از مشخصات تک تک اعضا جامعه آگاه است.

توجه: عمل تعویض متن، ما بین نخ‌های یک فرآیند (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در سطح هسته سیستم عامل، در مُد هسته پردازنده توسط زمان‌بند چند نخی در سطح هسته سیستم عامل انجام می‌گردد و عمل تعویض متن ما بین فرآیندها (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در سطح هسته سیستم عامل و در مُد هسته پردازنده توسط زمان‌بند پردازنده انجام می‌گردد.

توجه: به دلیل آنکه نخ‌های یک فرآیند (فرآیندهای کاربر یا فرآیندهای سیستم عامل)، تماماً در فضای هسته سیستم عامل مدیریت می‌شوند، و هسته سیستم عامل از وجود نخ‌های یک فرآیند آگاه است، اگر نخ موجود در یک فرآیند، یک فراخوان سیستمی مسدود کننده را اجرا نماید، هسته سیستم عامل فقط آن نخ مربوطه را مسدود می‌کند و نخ‌های دیگر هم خانواده با آن نخ مسدود شده، همچنان می‌توانند از پردازنده بهره ببرند.

توجه: این راه کار، منجر به امکان هم روندی (در سیستم‌های تک پردازنده‌ای) و امکان توازی (در سیستم‌های چند پردازنده‌ای) میان فرآیندهای مختلف و یا میان نخ‌های داخل یک فرآیند می‌شود. البته اگر نخ مسدود نگردد، درجه هم‌روندی و توازی بالاتر هم خواهد رفت، زیرا در این صورت همه نخ‌ها به طور هم روندی یا موازی در حال حرکت هستند. مانند یک تیم فوتبال ۱۱ نفره که اگر بازیکنی

مرتکب خطا گردد، همروندی یا توازی همچنان برقرار است، چون داور تک تک بازیکنان را می‌شناسد و فقط بازیکن خاطی را مسدود، جریمه و اخراج می‌کند و بقیه بازیکنان تیم به بازی خود ادامه می‌دهند، اما اگر هیچ یک از بازیکنان تیم مرتکب خطا نگردند، واضح است که همروندی و توازی بالاتر هم خواهد بود.

توجه: مثال کیک مطرح شده را مجدداً به یاد آورید، در روش سطح هسته، یک فرآیند چند نخ‌ی (فرآیندهای کاربر یا فرآیندهای سیستم عامل) می‌تواند از امتیازات چند پردازنده‌ای بهره‌بردار. در روش سطح هسته، هسته سیستم عامل می‌تواند در هر لحظه چندین پردازنده را در اختیار نخ‌های یک فرآیند قرار دهد. یعنی در این روش خوردن کیک بخش بندی شده به صورت چند نفری امکان‌پذیر است. بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند وجود دارد.

توجه: در این روش، منابع به فرآیندها اختصاص می‌یابند ولی زمان‌بندی پردازنده، بر روی نخ‌ها انجام می‌گیرد و زمان‌بندی نخ‌های سطح کاربر و نخ‌های سطح هسته، بر عهده زمان‌بند چند نخ‌ی سطح هسته می‌باشد.

توجه: سیستم عامل لینوکس، خانواده سیستم عامل ویندوز و سولاریس ۹ مدل یک به یک را پیاده‌سازی می‌کنند.

۳- روش ترکیبی (سطح کاربر و هسته) یا مدل چند به چند (many to many)

این روش از اجتماع دو روش سطح کاربر و هسته ابداع گردیده است. در این روش علاوه بر زمان‌بند پردازنده و زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل، زمان‌بند چند نخ‌ی نیز در سطح کاربر برای زمان‌بندی و اولویت‌دهی نخ‌های فرآیند کاربر وجود دارد. در واقع زمان‌بندی نخ‌های فرآیندهای سیستم عامل، توسط هسته سیستم عامل و زمان‌بندی نخ‌های فرآیندهای کاربر، توسط برنامه کاربر انجام می‌گردد، که این امر منجر به اولویت‌بندی نخ‌های فرآیندهای کاربر می‌گردد. در این روش، زمان‌بند چند نخ‌ی در سطح کاربر، نخ‌های کاندید خود را از میان نخ‌های متعدد در فرآیندهای مختلف کاربر بر اساس یک الگوریتم خاص انتخاب و تحویل زمان‌بند چند نخ‌ی در سطح هسته می‌دهد. در واقع انتخاب نخ‌های کاندید (زمان‌بندی) در فرآیندهای کاربر به خود کاربر واگذار شده است که همانطور که گفتیم منجر به اولویت‌بندی نخ‌های فرآیندهای کاربر می‌گردد. حال اجتماع حاصل از نخ‌های کاندید فرآیندهای کاربر و نخ‌های فرآیندهای سیستم عامل توسط زمان‌بند چند نخ‌ی هسته سیستم عامل بر اساس یک الگوریتم خاص زمان‌بندی می‌شود.

از توضیحات فوق این مفهوم برداشت می‌شود که هسته سیستم عامل باید این قابلیت را داشته باشد که کاربر بتواند نخ‌های فرآیندهای سطح خود را به هسته سیستم عامل معرفی کند. بنابراین در این روش هسته سیستم عامل نه تنها فرآیندهای کاربر را می‌شناسد، بلکه از وجود نخ‌های کاربر داخل فرآیندهای

کاربر نیز در صورت وجود نخ آگاه است.

توجه: برای معرفی نخ‌های کاندید فرآیندهای سطح کاربر به زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل، علاوه بر جایگاه‌های مخصوص نخ‌های فرآیندهای سیستم عامل در زمان‌بند چندنخی در سطح هسته سیستم عامل، تعدادی جایگاه، ویژه نخ‌های سطح کاربر نیز، در زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل در نظر گرفته شده است، به این جایگاه ویژه که **محیط اجرای نخ** نیز نامیده می‌شود، **LWP** گفته می‌شود. در واقع هر نخ کاندید انتخاب شده توسط زمان‌بند چند نخ‌ی در سطح کاربر، پس از معرفی به زمان‌بند چند نخ‌ی در سطح هسته، در یکی از جایگاه‌های ویژه که همان **LWP** است، جهت زمان‌بندی توسط زمان‌بند چند نخ‌ی در سطح هسته، قرار می‌گیرد.

توجه: **LWP** سرواژه عبارت **Light Weight Process** و به معنی فرآیند سبک وزن است.

توجه: هنگامی که یک نخ به زمان‌بند چند نخ‌ی در سطح هسته معرفی می‌گردد و در ادامه در یک **LWP** جهت زمان‌بندی توسط زمان‌بند چند نخ‌ی در سطح هسته، قرار می‌گیرد، در طول حیات خود ممکن است، در **LWP**‌های متفاوتی بخش‌هایی از اجرای خود را طی کند، مثلاً یک نخ در صورت رسیدن به عملیات ورودی و خروجی، جایگاه خود یعنی **LWP** را واگذار می‌کند و در اجرای بعدی پس از پایان عملیات ورودی و خروجی ممکن است به یک **LWP** دیگر منتسب شود. توجه کنید که **LWP محیط اجرای نخ** می‌باشد.

توجه: برای انجام زمان‌بندی چند نخ‌ی، هسته سیستم عامل باید علاوه بر جدول فرآیندها، یک جدول نخ (شبه جدول فرآیند) داشته باشد که اطلاعات تمامی نخ‌های موجود در سیستم را نگهداری کند. **توجه:** در این روش نخ ماهیت فیزیکی دارد و از دید کاربر و سیستم عامل وجود دارد، بنابراین در این روش، هر نخ، رجیستر و پشته مختص به خود را دارد، به عبارت دیگر هر نخ **TCB** مختص به خود را دارد.

توجه: عملیات مرتبط با نخ‌های سطح کاربر، مانند ایجاد (بارگذاری **TCB** مختص به نخ) و پایان دادن (ذخیره‌سازی **TCB** مختص به نخ) بر عهده هسته سیستم عامل است و توسط زمان‌بند چند نخ‌ی سطح هسته انجام می‌گردد.

توجه: مدل غیر کامپیوتری این روش نیز وجود دارد، مانند حالتی که درآمدهای دولت حاصل از منابع کشور، با اعمال نوعی اولویت‌بندی بین برخی از اعضای یک کشور تقسیم گردد (حال این اعضا شهروند عام باشند یا خاص) در این روش هر یک از اعضای اولویت‌دار و منتخب کشور شماره حساب مختص به خود را دارند. اما اگر یکی از اعضای خانواده خطایی انجام دهد و محکوم گردد، آنگاه فقط همان فرد برای مدتی از خدمات دولت محروم می‌گردد و دولت به بقیه اعضای خانواده آن فرد، همچنان خدمات ارائه می‌دهد، زیرا دولت از مشخصات تک تک اعضای آن خانواده آگاه است.

توجه: عمل تعویض متن مابین نخ‌های یک فرآیند (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در

سطح هسته سیستم عامل، در مُد هسته پردازنده توسط زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل انجام می‌گردد و عمل تعویض متن ما بین فرآیندها (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در سطح هسته سیستم عامل و در مُد هسته پردازنده توسط زمان‌بند پردازنده انجام می‌گردد.

توجه: به دلیل آنکه نخ‌های کاندید فرآیندهای سطح کاربر، به هسته سیستم عامل معرفی می‌گردند و هسته سیستم عامل از وجود نخ‌های کاندید فرآیندهای سطح کاربر آگاه است، اگر یک نخ کاندید موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدود کننده را اجرا نماید، هسته سیستم عامل فقط آن نخ مربوطه را مسدود می‌کند و نخ‌های دیگر هم خانواده با آن نخ مسدود شده، همچنان می‌توانند از پردازنده بهره ببرند. این راه کار، منجر به امکان هم روندی (در سیستم‌های تک پردازنده‌ای) و امکان توازی (در سیستم‌های چند پردازنده‌ای) میان فرآیندهای مختلف کاربر و یا میان نخ‌های داخل یک فرآیند کاربر می‌شود. البته اگر نخ‌ی مسدود نگردد، درجه هم روندی و توازی بالاتر هم خواهد رفت، زیرا در این صورت همه نخ‌ها به طور هم روند یا موازی در حال حرکت هستند.

مانند یک تیم فوتبال ۱۱ نفره که اگر بازیکنی مرتکب خطا گردد، هم‌روندی یا توازی همچنان برقرار است، چون داور تک تک بازیکنان را می‌شناسد و فقط بازیکن خاطی را مسدود، جریمه و اخراج می‌کند و بقیه بازیکنان تیم به بازی خود ادامه می‌دهند، اما اگر هیچ یک از بازیکنان تیم مرتکب خطا نگردند، واضح است که هم‌روندی و توازی بالاتر هم خواهد بود.

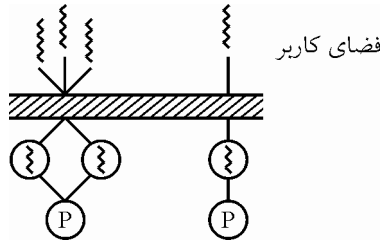
توجه: همانند نخ‌های سطح کاربر، هم روندی و توازی برای نخ‌های سطح هسته سیستم عامل نیز، در حالت انسداد یک نخ یا عدم انسداد یک نخ، برقرار است.

توجه: مثال کیک مطرح شده را مجدداً به یاد آورید، در روش ترکیبی، یک فرآیند چند نخ‌ی (فرآیندهای کاربر یا فرآیندهای سیستم عامل) می‌تواند از امتیازات چند پردازنده‌ای بهره ببرد. در روش ترکیبی، هسته سیستم عامل می‌تواند در هر لحظه چندین پردازنده را در اختیار نخ‌های یک فرآیند قرار دهد. یعنی در این روش خوردن کیک بخش بندی شده به صورت چند نفری امکان‌پذیر است. بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند وجود دارد.

توجه: در این روش نیز، منابع به فرآیندها اختصاص می‌یابند ولی زمان‌بندی پردازنده، بر روی نخ‌ها انجام می‌گیرد. اما زمان‌بندی نخ‌های سطح کاربر، بر عهده زمان‌بند چند نخ‌ی سطح کاربر و زمان‌بندی نخ‌های سطح هسته، بر عهده زمان‌بند چند نخ‌ی سطح هسته خواهد بود.

تست‌های فصل سوم: مدیریت نخ

۱- در یک سیستم کامپیوتری نحوه استفاده از نخ (Thread) در لایه کاربر و در لایه کرنل به صورت مقابل نشان داده شده است. کدام عبارت صحیح است؟ (مهندسی کامپیوتر - دولتی ۸۹)



- فرآیند P
 } نخ در لایه کاربر
 } نخ در لایه کرنل

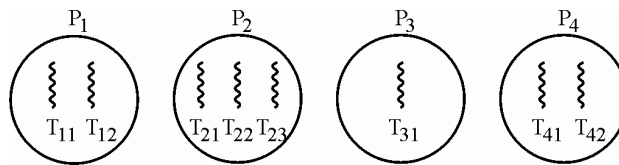
۱) فراخوانی‌های سیستمی از نوع مسدود (Blocking) با تأمین همروندی حمایت می‌شوند و برای فراخوانی‌های سیستمی از نوع غیرمسدود (Non-Blocking) درجه همروندی پایین‌تر است.

۲) فراخوانی‌های سیستمی از نوع مسدود (Blocking) بدون تأمین همروندی اجرا می‌شوند و فراخوانی‌های سیستمی از نوع غیرمسدود (Non-Blocking) همروندی را تأمین می‌کنند.

۳) فراخوانی‌های سیستمی از نوع مسدود (Blocking)، با تأمین همروندی حمایت می‌شوند و برای فراخوانی‌های سیستمی از نوع غیرمسدود (Non-Blocking) درجه همروندی بالاتر است.

۴) فراخوانی‌های سیستمی از نوع مسدود (Blocking) بدون تأمین همروندی اجرا می‌شوند و فراخوانی‌های سیستمی از نوع غیرمسدود (Non-Blocking) نیز با مشکل همروندی مواجه هستند.

۲- سیستمی شامل 4 فرآیند است که داخل هر فرآیند می‌تواند بیش از یک نخ (Thread) اجرایی وجود داشته باشد. در لحظه صفر وضعیت این 4 فرآیند و تعداد نخ‌های اجرایی آنها در شکل و جدول زیر مشخص شده است؟



زمان لازم برای اجرای نخ‌ها

فرآیند	P ₁		P ₂			P ₃	P ₄	
نخ	T ₁₁	T ₁₂	T ₂₁	T ₂₂	T ₂₃	T ₃₁	T ₄₁	T ₄₂
زمان اجرا (ms)	12	9	7	8	8	9	7	8

سه‌م زمانی هر فرآیند 10ms است و از روش نوبت‌گردشی (RR) استفاده می‌شود. همچنین داخل هر فرآیند از روش FIFO برای تعویض نخ‌ها استفاده می‌شود و تا زمان اجرایی یک نخ تمام نشده نوبت به نخ بعدی نمی‌رسد. برای تعویض فرآیند 1ms و برای تعویض نخ در داخل فرآیند 0.5ms زمان لازم است. زمان پایان نخ‌های T_{12} و T_{22} چقدر است؟ (مهندسی IT - دولتی ۸۹)

- (۱) 73ms و 59.5ms
(۲) 70ms و 45.5ms
(۳) 45.5ms و 70.5ms
(۴) 18ms و 29ms

۳- در این سوال تفاوت‌های مابین تغییر متن (context switch) در فرآیندها و تغییر متن در نخ‌های یک فرآیند بررسی می‌شوند. کدام گزینه صحیح است؟ (مهندسی کامپیوتر - دولتی ۹۰)

- (۱) تغییر متن در فرآیندها موجب تغییر ثبات‌های اجرایی برنامه می‌شود و سیاست‌های حفاظتی را تغییر می‌دهد.
تغییر متن در نخ‌ها موجب تغییر اشاره‌گر پشته (SP) و پاک شدن TLB می‌شود.
(۲) تغییر متن در فرآیندها موجب تغییر اشاره‌گر پشته (SP) و پاک شدن TLB می‌شود.
تغییر متن در نخ‌ها موجب تخصیص سهم زمانی تازه می‌شود و برنامه‌شمار (PC) تغییر می‌یابد.
(۳) تغییر متن در فرآیندها موجب تغییر ثبات‌های اجرایی برنامه می‌شود و با TLB کاری ندارد.
تغییر متن در نخ‌ها موجب تغییر اشاره‌گر پشته (SP) و پاک شدن TLB می‌شود.
(۴) تغییر متن در فرآیندها موجب تغییر ثبات‌های اجرایی برنامه و تغییر برنامه‌شماره (PC) می‌شود.
تغییر متن در نخ‌ها، ثبات‌ها و جداول مدیریت حافظه را تغییر نمی‌دهد و اشاره‌گر پشته (SP) را تغییر می‌دهد.

۴- در رابطه با مدیریت نخ (thread) کدام یک از جملات زیر صحیح است؟ (مهندسی IT - دولتی ۹۱)
(توجه: LWP مخفف Light Weight Process است و محیط اجرایی نخ می‌باشد)

- (۱) تغییر متن (Context switch) ما بین نخ‌های شامل: (۱) ذخیره ثبات‌های پردازنده مربوط به نخ بیرون رونده و بار کردن ثبات‌های پردازنده مربوط به نخ داخل شونده و (۲) ذخیره لیست فایل‌های باز شده توسط نخ است.
(۲) یک نخ عادی در طول حیات خود ممکن است در LWP های متفاوتی بخش‌هایی از اجرای خود را بگذارد.
(۳) به ازای هر نخ سیستم عامل یک LWP ایجاد می‌کند و نخ تا پایان حیات خود به آن LWP منتسب است، زمان‌بندی نخ می‌تواند توسط سیستم عامل یا کاربر انجام پذیرد.
(۴) نخ مستقیماً زیر نظر سیستم عامل اجرا می‌شود و مدیریت آن نمی‌تواند در سطح کاربر باشد.

- ۵- کدام گزینه درباره مدل‌های چند نخ‌ی درست نیست؟ (مهندسی کامپیوتر - دولتی ۹۶)
- (۱) مدل‌های یک به یک و چند به چند توانایی بکارگیری بهتر از پردازنده‌ها / هسته‌ها را دارند.
- (۲) مدل چند به یک نسبت به مدل یک به یک از کارایی کمتری برخوردار است.
- (۳) در مدل‌های یک به یک و چند به یک امکان همزمانی کامل بین نخ‌ها وجود دارد.
- (۴) مدل یک به یک نسبت به مدل چند به یک از همزمانی بیشتری برخوردار است.

- ۶- اگر در یک پردازنده دو هسته‌ای از زمان‌های انتظار نخ‌ها (Threads) برای عملیات حافظه (Memory Stall) جهت سیکل محاسباتی سایر نخ‌ها استفاده شود. بهترین ترکیب تخصیص چهار نخ زیر به هسته‌ها برای کاهش زمان تأخیر چگونه خواهد بود؟ (مهندسی IT - دولتی ۹۶)

T_1	C	M	C	M
	4	3	7	3

توضیح: سیکل محاسباتی هر نخ با C و سیکل حافظه با M نشان داده شده و مدت زمان هر سیکل به ثانیه، زیر آن نشان داده شده است. پردازنده، اجرای سیکل C را تا زمانی ادامه می‌دهد که به سیکل M برسد و پس از آن سیکل C نخ دیگری را اجرا می‌نماید.

T_2	C	M	C	M
	5	2	6	2

T_3	C	M	C	M
	2	6	4	6

- (۱) T_2 و T_1 روی هسته اول، بقیه روی هسته دوم
- (۲) T_1 و T_3 روی هسته اول، بقیه روی هسته دوم
- (۳) T_1 و T_4 روی هسته اول، بقیه روی هسته دوم
- (۴) T_2 و T_4 روی هسته اول، بقیه روی هسته دوم

T_4	C	M	C	M
	2	2	2	2

- ۷- برای یک نخ (Thread) کدام یک از تغییر وضعیت‌ها، امکان‌پذیر است؟ (مهندسی IT - دولتی ۹۶)
- (۱) رفتن از حالت آماده به خاتمه
- (۲) رفتن از حالت آماده به انتظار
- (۳) رفتن از حالت آماده اجرا به انتظار
- (۴) رفتن از حالت آماده انتظار به اجرا

پاسخ تست‌های فصل سوم: مدیریت نخ

۱- گزینه (۳) صحیح است.

به طور کلی مدیریت و زمان‌بندی نخ‌ها به سه روش زیر انجام می‌گیرد:

۱- روش سطح کاربر یا مدل چند به یک (many to one)

۲- روش سطح هسته یا مدل یک به یک (one to one)

۳- روش ترکیبی (سطح کاربر و هسته) یا مدل چند به چند (many to many)

با توجه به شکل مطرح شده در صورت سؤال، واضح است که در این سیستم کامپیوتری مدیریت و زمان‌بندی نخ‌ها به روش ترکیبی انجام می‌گردد.

در روش ترکیبی به دلیل آنکه نخ‌های کاندید فرآیندهای سطح کاربر، به هسته سیستم عامل معرفی می‌گردند و هسته سیستم عامل از وجود نخ‌های کاندید فرآیندهای سطح کاربر آگاه است، اگر یک نخ کاندید موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدود کننده را اجرا نماید، هسته سیستم عامل فقط آن نخ مربوطه را مسدود می‌کند و نخ‌های دیگر هم خانواده با آن نخ مسدود شده، همچنان می‌توانند از پردازنده بهره ببرند. این راه کار، منجر به امکان هم‌روندی (در سیستم‌های تک پردازنده‌ای) و امکان توازی (در سیستم‌های چند پردازنده‌ای) میان فرآیندهای مختلف کاربر و یا میان نخ‌های داخل یک فرآیند کاربر می‌شود. البته اگر نخی مسدود نگردد، درجه هم‌روندی و توازی بالاتر هم خواهد رفت، زیرا در این صورت همه نخ‌ها بطور هم‌روند یا موازی در حال حرکت هستند. مانند یک تیم فوتبال ۱۱ نفره که اگر بازیکنی مرتکب خطا گردد، هم‌روندی یا توازی همچنان برقرار است، چون داور تک تک بازیکنان را می‌شناسد و فقط بازیکن خاطی را اخراج، جریمه و مسدود می‌کند و بقیه بازیکنان تیم به بازی خود ادامه می‌دهند، اما اگر هیچ یک از بازیکنان تیم مرتکب خطا نگردند، واضح است که هم‌روندی و توازی بالاتر هم خواهد بود.

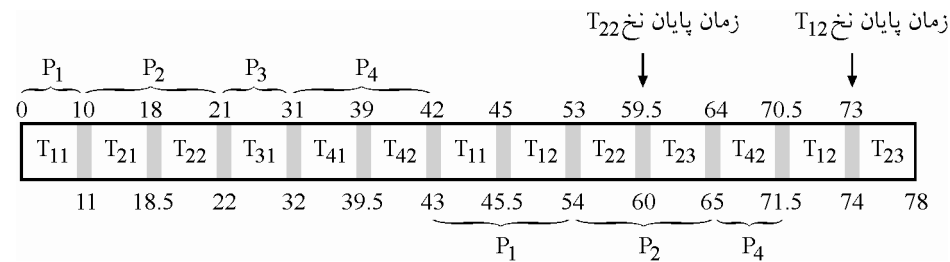
توجه: همانند نخ‌های سطح کاربر، هم‌روندی و توازی برای نخ‌های سطح هسته نیز، برقرار است.

۲- گزینه (۱) صحیح است.

با توجه به جدول مطرح شده در صورت سؤال داریم:

فرآیند	نخ	زمان اجرای نخ
P ₁	T ₁₁	12
	T ₁₂	9
P ₂	T ₂₁	7
	T ₂₂	8
	T ₂₃	8
P ₃	T ₃₁	9
P ₄	T ₄₁	7
	T ₄₂	8

با توجه به اعمال الگوریتم RR با $Q=10$ در سطح فرآیندها و به کارگیری FCFS در سطح نخ‌ها، نمودار زمانی گانت را ترسیم می‌کنیم.



زمان پایان نخ T_{12} برابر 73 و زمان پایان نخ T_{22} برابر 59.5 می‌باشد.

۳- گزینه (۴) صحیح است.

یک فرآیند در واقع شامل اجزایی مانند رجیسترهای پردازنده (از جمله شمارنده برنامه (PC)، یک پشته در حافظه (جهت نگهداری داده‌های موقتی، متغیرهای محلی، پارامترهای توابع و زیر روال‌ها) و بخش داده‌ای (Data Segment) (شامل داده‌ها و متغیرهای سراسری) است. جهت نگهداری اطلاعات مربوط به فرآیندها نزد سیستم عامل از PCB استفاده می‌گردد، محتوای PCB برای هر فرآیند به صورت زیر است:

- شناسه فرآیند (Process ID)
- شناسه کاربر فرآیند یا حتی شناسه گروهی که کاربر فرآیند عضو آن است. (UID و GID)
- اولویت فرآیند
- رجیسترهای پردازنده (شامل PC, PSW و مابقی رجیسترها)
- حالت و وضعیت فرآیند (مانند جدید، آماده، در حال اجرا، منتظر و معلق)
- اطلاعات راجع به منابع در اختیار فرآیند (فایل‌ها، حافظه، پردازنده)
- اطلاعاتی راجع به میزان مصرف منابع توسط فرآیند (مانند میزان پردازنده مصرفی)
- اطلاعات مربوط به زمان‌بندی
- اشاره‌گرهایی به قسمت کد، داده و پشته (SP, DS, CS)
- اطلاعات مربوط به مدیریت حافظه (جداول صفحه، قطعه و مسائل حفاظتی)

توجه: در واقع PCB مانند یک مخزن است که اطلاعاتی که از یک فرآیند به فرآیند دیگر متغیر است، در آن ذخیره می‌شود.

پرونده حیاتی یک فرآیند در طول حیاتش در یک ساختمان داده به نام PCB نگهداری می‌گردد، هر فرآیند قبل از تعویض متن، پرونده حیاتی خود را در PCB قرار می‌دهد تا هنگام بازگشت مجدد، برای ادامه کار، این پرونده حیاتی بتواند مورد استفاده پردازنده و خود فرآیند قرار گیرد. زیرا پردازنده قادر به نگهداری اطلاعات مربوط به تمام فرآیندها نیست، چون در هر لحظه فقط

امکانات و فضای کار لازم برای اجرای یک فرآیند را دارد. بنابراین هر فرآیند بطور مستقل از سایر فرآیندها در طول حیات خود باید پرونده حیاتی خود را درون PCB نگهداری کند. پرونده حیاتی فرآیند، مانند پرونده پزشکی یک انسان در طول حیاتش می‌باشد. تا ادامه درمان انسان، در طول حیاتش امکان پذیر باشد. زیرا پزشکان قادر به نگهداری وضعیت درمانی همه بیماران در داخل ذهن خود نیستند.

توجه: نخ‌های همتا که در یک فرآیند قرار دارند، از کد، داده و منابع مشترک استفاده می‌کنند، اما هر نخ، شمارنده برنامه (PC)، مجموعه رجیستر و فضای پشته جداگانه‌ای در اختیار دارد. در واقع هر نخ، TBC مجزایی دارد.

گزینه اول: تغییر متن در فرآیندها منجر به تغییر ثبات‌ها، سیاست‌های حفاظتی و تغییر اشاره‌گر پشته (SP) می‌گردد. اما تغییر متن نخ‌ها منجر به پاک شدن TLB نمی‌گردد، توجه کنید که TLB برای ترجمه سریع صفحات یک فرآیند مورد استفاده قرار می‌گیرد. بنابراین گزینه اول نادرست است.

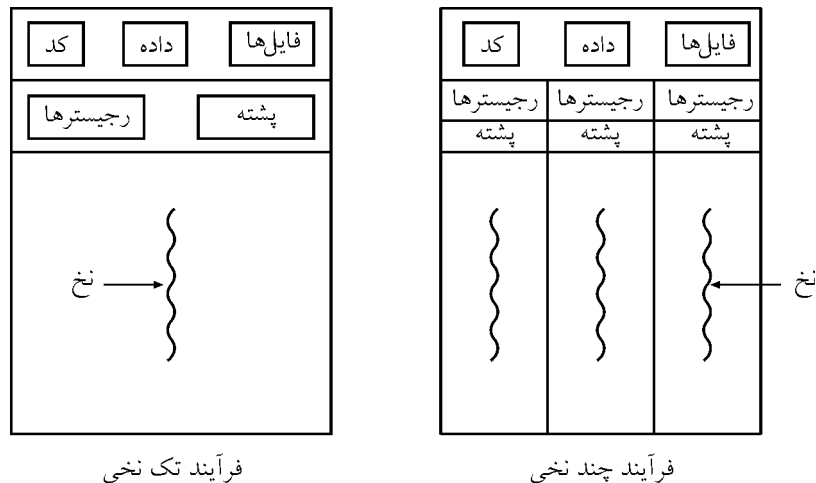
گزینه دوم: تغییر متن در فرآیند منجر به تغییر اشاره‌گر پشته، تغییر برنامه شمار (PC) و پاک شدن TLB می‌گردد، جدول TLB براساس فرآیند در حال اجرا پر می‌شود. تغییر متن در نخ‌ها همیشه باعث تخصیص سهم زمانی تازه نمی‌شود. چون سهم زمانی موقعی به نخ تعلق می‌گیرد که الگوریتم زمان‌بندی نخ‌ها RR باشد و اگر FIFO باشد سهم زمانی تازه معنا ندارد، سهمی در کار نیست. بنابراین گزینه دوم نادرست است.

گزینه سوم: تغییر متن در فرآیندها منجر به تغییر ثبات‌ها و پاک شدن TLB می‌گردد. جدول TLB براساس فرآیند در حال اجرا پر می‌شود. همچنین تغییر متن در نخ‌ها منجر به تغییر اشاره‌گر پشته (SP) می‌گردد، اما با TLB کاری ندارد. بنابراین گزینه سوم نادرست است.

گزینه چهارم: تغییر متن در فرآیندها منجر به تغییر ثبات‌های اجرایی برنامه و تغییر برنامه شمار (PC) می‌گردد. تغییر متن در نخ‌ها منجر به تغییر ثبات‌ها و جداول مربوط به مدیریت حافظه نمی‌گردد. اما منجر به تغییر اشاره‌گر (SP) می‌گردد. بنابراین گزینه چهارم درست است.

۴- گزینه (۲) صحیح است.

گزینه اول نادرست است. زیرا عبارت دوم نادرست، چون لیست فایل‌های باز شده، برای کل یک فرآیند ذخیره می‌شود. لذا در عمل تعویض متن بین نخ‌ها، لیست فایل‌های باز شده، توسط نخ ذخیره نمی‌شود. در واقع فایل‌های باز شده، برای فرآیند است و به شکل مشترک نخ‌های آن فرآیند، از آن استفاده می‌کنند. عبارت اول گزینه اول درست است، زیرا هر نخ ثبات‌های مخصوص خود را دارد که در پی عمل تعویض متن باید ثبات‌های خود را برای استفاده در آینده ذخیره کند. اما نخ وارد شونده، باید ثبات‌های خود را برای ادامه کار، باز کند. به شکل زیر توجه کنید:



فرآیند تک نخ

فرآیند چند نخ

گزینه دوم درست است زیرا:

بطور کلی مدیریت و زمان‌بندی نخ‌ها به سه روش زیر انجام می‌گیرد:

۱- روش سطح کاربر یا مدل چند به یک (many to one)

۲- روش سطح هسته یا مدل یک به یک (one to one)

۳- روش ترکیبی (سطح کاربر و هسته) یا مدل چند به چند (many to many)

در روش ترکیبی موارد زیر برقرار است:

توجه: برای معرفی نخ‌های کاندید فرآیندهای سطح کاربر به زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل، علاوه بر جایگاه‌های مخصوص نخ‌های فرآیندهای سیستم عامل در زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل، تعدادی جایگاه، ویژه نخ‌های سطح کاربر نیز در زمان‌بند چند نخ‌ی در سطح هسته سیستم عامل در نظر گرفته شده است، به این جایگاه ویژه که محیط اجرای نخ نیز نامیده می‌شود، LWP گفته می‌شود. در واقع هر نخ کاندید انتخاب شده توسط زمان‌بند چند نخ‌ی در سطح کاربر، پس از معرفی به زمان‌بند چند نخ‌ی در سطح هسته، در یکی از جایگاه‌های ویژه که همان LWP است، جهت زمان‌بندی توسط زمان‌بند چند نخ‌ی در سطح هسته، قرار می‌گیرد.

توجه: LWP سرواژه عبارت Light Weight Process و به معنی فرآیند سبک وزن است.

توجه: هنگامی که یک نخ به زمان‌بند چند نخ‌ی در سطح هسته معرفی می‌گردد و در ادامه در یک LWP جهت زمان‌بندی توسط زمان‌بند چند نخ‌ی در سطح هسته قرار می‌گیرد، در طول حیات خود ممکن است در LWP‌های متفاوتی بخش‌هایی از اجرای خود را طی کند، مثلاً یک نخ در صورت رسیدن به عملیات ورودی و خروجی، جایگاه خود یعنی LWP را واگذار می‌کند و در اجرای بعدی پس از پایان عملیات ورودی و خروجی ممکن است به یک LWP دیگر منتسب شود. توجه کنید که LWP محیط اجرای نخ می‌باشد. مانند بچه‌ای که داخل یکی از کابین‌های یک

چرخ و فلک است، اما پس از مدتی به دلیل مسائل ورودی و خروجی، مجبور به ترک کابین چرخ و فلک می‌شود، اما بعدها پس از حل مسائل ورودی و خروجی ممکن است در سوار شدن مجدد در چرخ و فلک کابین دیگری را ملاقات کند.

گزینه سوم نادرست است، زیرا عبارت اول این گزینه براساس مطالب فوق نادرست است. اما عبارت دوم این گزینه درست است، زیرا زمان‌بندی نخ مطابق مطالب فوق می‌تواند توسط سیستم عامل یا کاربر و یا حتی هر دو توأم با هم و به شکل ترکیبی انجام گردد. گزینه چهارم نادرست است، مطابق آنچه برای گزینه سوم شرح داده شد.

۵- گزینه (۳) صحیح است.

به طور کلی مدیریت و زمان‌بندی نخ‌ها به سه روش زیر انجام می‌گردد:

۱- روش سطح کاربر یا مدل چند به یک (many to one)

در این روش فقط زمان‌بند پردازنده و زمان‌بند چند نخ‌ی در سطح کاربر وجود دارد و زمان‌بند چند نخ‌ی در سطح هسته در این روش مورد استفاده قرار نمی‌گیرد. در واقع هسته سیستم عامل فقط فرآیندها را می‌شناسد و هیچ اطلاعاتی از نخ‌ها ندارد.

توجه: به دلیل آنکه نخ‌های یک فرآیند کاربر، تماماً در فضای کاربر مدیریت می‌شوند، اگر نخ موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدودکننده را اجرا نماید، هسته سیستم عامل نه تنها آن نخ، بلکه کل فرآیند کاربر را که شامل تمام نخ‌های دیگر می‌باشد، مسدود می‌کند، زیرا هسته سیستم عامل خبری از نخ‌های داخل فرآیند کاربر ندارد، در واقع در این روش هسته سیستم عامل اصلاً زمان‌بند چند نخ‌ی ندارد که بخواهد، چند نخ‌ی را زمان‌بندی کند!

این راه‌کار، منجر به عدم امکان هم‌روندی (در سیستم‌های تک پردازنده‌ای) نخ‌های داخل یک فرآیند کاربر در حالت انسداد یک نخ داخل یک فرآیند کاربر می‌گردد. البته اگر نخ‌ی داخل یک فرآیند کاربرد مسدود نگردد، امکان هم‌روندی میان نخ‌های داخل فرآیند کاربر برقرار است.

توجه: در روش سطح کاربر، یک فرآیند چند نخ‌ی کاربر نمی‌تواند از امتیازات چند پردازنده‌ای بهره‌بردارد. زیرا در روش سطح کاربر، هسته سیستم عامل در هر لحظه فقط یک پردازنده را در اختیار نخ‌های یک فرآیند کاربر قرار می‌دهد. حتی اگر چند پردازنده موجود باشد.

بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند کاربر وجود ندارد.

۲- روش سطح هسته یا مدل یک به یک (one to one)

در این روش فقط زمان‌بند پردازنده و زمان‌بند چند نخ‌ی در سطح هسته وجود دارد و زمان‌بند چند نخ‌ی در سطح کاربر در این روش مورد استفاده قرار نمی‌گیرد. در واقع هسته سیستم عامل نه تنها فرآیندها را می‌شناسد بلکه از وجود نخ‌های داخل یک فرآیند نیز در صورت وجود نخ آگاه است.

توجه: به دلیل آنکه نخ‌های یک فرآیند (فرآیندهای کاربر یا فرآیندهای سیستم عامل)، تماماً در فضای هسته سیستم عامل مدیریت می‌شوند، و هسته سیستم عامل از وجود نخ‌های یک فرآیند آگاه است، اگر نخ موجود در یک فرآیند، یک فراخوان سیستمی مسدودکننده را اجرا نماید، هسته سیستم عامل فقط آن نخ مربوطه را مسدود می‌کند و نخ‌های دیگر هم خانواده با آن نخ مسدود شده، همچنان می‌توانند از پردازنده بهره ببرند.

توجه: این راه‌کار، منجر به امکان هم‌روندی (در سیستم‌های تک پردازنده‌ای) و امکان توازی (در سیستم‌های چند پردازنده‌ای) میان فرآیندهای مختلف و یا میان نخ‌های داخل یک فرآیند می‌شود. البته اگر نخ مسدود نگردد، درجه هم‌روندی و توازی بالاتر هم خواهد رفت، زیرا در این صورت همه نخ‌ها به طور هم‌روند یا موازی در حال حرکت هستند.

توجه: در روش سطح هسته، یک فرآیند چند نخ (فرآیندهای کاربر یا فرآیندهای سیستم عامل) می‌تواند از امتیازات چند پردازنده‌ای بهره ببرد. در روش سطح هسته، هسته سیستم عامل می‌تواند در هر لحظه چندین پردازنده را در اختیار نخ‌های یک فرآیند قرار دهد. بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند وجود دارد.

۳- روش ترکیبی (سطح کاربر و هسته) یا مدل چند به چند (many to many)

این روش از اجتماع دو روش سطح کاربر و هسته ابداع گردیده است. در این روش علاوه بر زمان‌بند پردازنده و زمان‌بند چند نخ در سطح هسته سیستم عامل، زمان‌بند چند نخ نیز در سطح کاربر برای زمان‌بندی و اولویت‌دهی نخ‌های فرآیند کاربر وجود دارد. در واقع زمان‌بندی نخ‌های فرآیندهای سیستم عامل، توسط هسته سیستم عامل و زمان‌بندی نخ‌های فرآیندهای کاربر، توسط برنامه کاربر انجام می‌گردد، که این امر منجر به اولویت‌بندی نخ‌های فرآیندهای کاربر می‌گردد.

توجه: به دلیل آنکه نخ‌های کاندید فرآیندهای سطح کاربر، به هسته سیستم عامل معرفی می‌گردند و هسته سیستم عامل از وجود نخ‌های کاندید فرآیندهای سطح کاربر آگاه است، اگر یک نخ کاندید موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدودکننده را اجرا نماید، هسته سیستم عامل فقط آن نخ مربوطه را مسدود می‌کند و نخ‌های دیگر هم خانواده با آن نخ مسدود شده، همچنان می‌توانند از پردازنده بهره ببرند.

توجه: این راه‌کار، منجر به امکان هم‌روندی (در سیستم‌های تک پردازنده‌ای) و امکان توازی (در سیستم‌های چند پردازنده‌ای) میان فرآیندهای مختلف کاربر و یا میان نخ‌های داخل یک فرآیند کاربر می‌شود. البته اگر نخ مسدود نگردد، درجه هم‌روندی و توازی بالاتر هم خواهد رفت، زیرا در این صورت همه نخ‌ها به طور هم‌روند یا موازی در حال حرکت هستند.

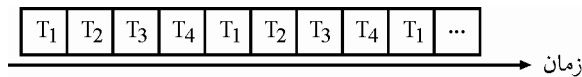
توجه: در روش ترکیبی، یک فرآیند چند نخ (فرآیندهای کاربر یا فرآیندهای سیستم عامل) می‌تواند از امتیازات چند پردازنده‌ای بهره ببرد. در روش ترکیبی، هسته سیستم عامل می‌تواند در هر لحظه چندین پردازنده را در اختیار نخ‌های یک فرآیند قرار دهد. بنابراین در این روش نیز

امکان پردازش موازی نخ‌های داخل یک فرآیند وجود دارد. مطابق مطالب بیان شده، گزینه‌های اول، دوم و چهارم درست هستند و گزینه سوم نادرست است. البته در گزینه سوم این عبارت که در مدل‌های یک به یک امکان همزمانی کامل بین نخ‌ها وجود دارد درست و این عبارت که در مدل‌های چند به یک امکان همزمانی کامل بین نخ‌ها وجود دارد نادرست است، زیرا در مدل چند به یک از آن جا که هسته سیستم عامل، نخ‌های کاربر را نمی‌شناسد، در صورت انسداد یک نخ کاربر، سیستم عامل کل فرآیند را مسدود می‌کند، بنابراین همزمانی و همروندی کامل در مدل چند به یک بین نخ‌های کاربر همواره برقرار نیست.

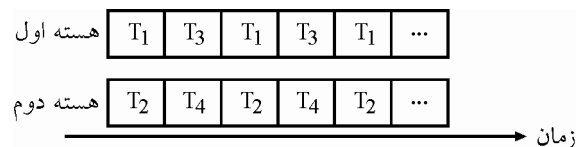
۶- گزینه (۱) صحیح است.

روند اخیر در طراحی سیستم‌ها، به گونه‌ای است که چندین هسته محاسباتی را در یک تراشه قرار می‌دهند. به گونه‌ای که هر هسته به صورت یک پردازنده مجزا برای سیستم عامل تلقی می‌شود. برنامه‌نویسی چند رشته نخ، مکانیزمی را برای استفاده مؤثرتر از هسته‌های سیستم فراهم نموده و توازی را افزایش می‌دهد. یک فرآیند را با چهار رشته نخ در نظر بگیرید. در یک سیستم با یک هسته پردازش، سیستم در یک زمان تنها یک رشته نخ را اجرا می‌کند. در یک سیستم چند رشته نخ، با چند هسته توازی به این معنی است که رشته نخ‌ها می‌توانند به طور موازی اجرا شوند و سیستم می‌تواند در هر هسته یک رشته نخ را قرار دهد.

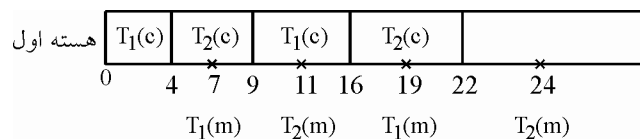
شکل زیر اجرای همروند یک فرآیند چند نخ را در یک سیستم تک هسته‌ای نمایش می‌دهد:



شکل زیر اجرای موازی یک فرآیند چند نخ را در یک سیستم دو هسته‌ای نمایش می‌دهد:



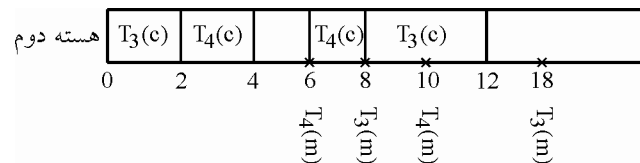
گزینه دوم: T_1 و T_3 روی هسته اول و T_2 و T_4 روی هسته دوم
گزینه چهارم: T_2 و T_4 روی هسته اول و T_1 و T_3 روی هسته دوم
 بنابراین گزینه دوم با گزینه چهارم فرقی ندارد!!!
 حال به بررسی نخ‌ها می‌پردازیم:
 اگر T_1 و T_2 روی هسته اول و T_3 و T_4 روی هسته دوم باشند:



بنابراین زمان خروج نخ‌های T_1 و T_2 به صورت زیر است:

T_1 خروج = 19 (3 ثانیه حافظه بعد از 16)

T_2 خروج = 24 (2 ثانیه حافظه بعد از 22)



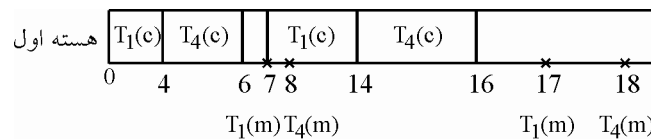
بنابراین زمان خروج نخ‌های T_3 و T_4 به صورت زیر است:

T_3 خروج = 18 (6 ثانیه حافظه بعد از 12)

T_4 خروج = 10 (2 ثانیه حافظه بعد از 8)

$$\text{متوسط زمان تأخیر} = \frac{T_1 + T_2 + T_3 + T_4}{4} = \frac{19 + 24 + 18 + 10}{4} = \frac{71}{4} = 17.75$$

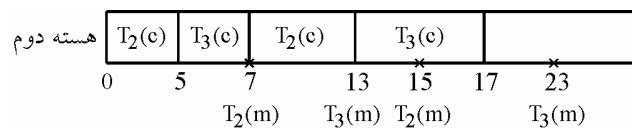
اگر T_1 و T_4 روی هسته اول و T_3 و T_2 روی هسته دوم باشند:



بنابراین زمان خروج نخ‌های T_1 و T_4 به صورت زیر است:

T_1 خروج = 17 (3 ثانیه حافظه پس از 14)

T_4 خروج = 18 (2 ثانیه حافظه پس از 16)



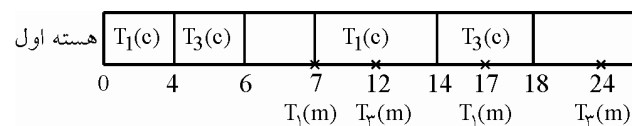
بنابراین زمان خروج نخ‌های T_2 و T_3 به صورت زیر است:

T_2 خروج = 15 (2 ثانیه حافظه پس از 13)

T_3 خروج = 23 (6 ثانیه حافظه پس از 17)

$$\text{متوسط زمان تأخیر} = \frac{T_1 + T_2 + T_3 + T_4}{4} = \frac{17 + 15 + 23 + 18}{4} = \frac{73}{4} = 18.25$$

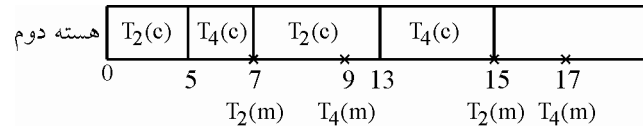
اگر T_1 و T_3 روی هسته اول و T_2 و T_4 روی هسته دوم باشند:



بنابراین زمان خروج نخ‌های T_1 و T_3 به صورت زیر است:

T_1 خروج = 17 (3 ثانیه حافظه بعد از 14)

T_3 خروج = 24 (6 ثانیه حافظه بعد از 18)



بنابراین زمان خروج نخ‌های T_2 و T_4 به صورت زیر است:

T_2 خروج = 15 (2 ثانیه حافظه بعد از 13)

T_4 خروج = 17 (2 ثانیه حافظه بعد از 15)

$$\text{متوسط زمان تأخیر} = \frac{T_1 + T_2 + T_3 + T_4}{4} = \frac{17 + 15 + 24 + 17}{4} = \frac{73}{4} = 18.25$$

۷- گزینه () صحیح است.

نخ‌ها هم مانند فرآیندها می‌توانند حالت‌های مختلفی را تجربه کنند مانند آماده، در حال اجرا یا منتظر (مسدود). در واقع پردازنده می‌تواند بین نخ‌ها به اشتراک گذاشته شود. حالت آماده اجرا همان معنی حالت آماده را می‌دهد بنابراین گزینه دوم و سوم یکسان هستند، سازمان سنجش آموزش کشور در کلید اولیه خود ابتدا گزینه سوم را به عنوان پاسخ اعلام کرده بود، سپس در کلید نهایی نظر خود را عوض کرد و گزینه سوم را با تأثیر مثبت اعلام کرد.

در روش سطح کاربر یا مدل چند به یک (Many to One) فقط زمان‌بند پردازنده و زمان‌بند چندنخی در سطح کاربر وجود دارد و زمان‌بند چند نخی در سطح هسته در این روش مورد استفاده قرار نمی‌گیرد. در واقع هسته سیستم عامل فقط فرآیندها را می‌شناسد و هیچ اطلاعاتی از نخ‌ها ندارد. در واقع اولویت‌بندی نخ‌ها، مدیریت نخ‌ها و زمان‌بند چندنخی در سطح کاربر و توسط یک بسته نرم‌افزاری انجام می‌گردد. بدین معنی که نخ‌ها را برنامه‌نویس مشخص می‌کند و مدیریت آن‌ها را نیز بر عهده می‌گیرد. بنابراین زمان‌بند پردازنده، براساس الگوریتم مشخصی مثلاً نوبت چرخشی پردازنده را در اختیار یکی از فرآیندهای آماده قرار می‌دهد. سپس زمان‌بند چند نخی در سطح کاربر، متناسب با کاربردی که در آن فرآیند به کار گرفته می‌شود، الگوریتم زمان‌بند را انتخاب کرده و تصمیم می‌گیرد که پردازنده در اختیار کدام یک از نخ‌های آماده در فرآیند موردنظر قرار گیرد و تا زمانی که پردازنده در تملک فرآیند باشد و یا تا قبل از پایان برش زمانی مربوط به فرآیند، نخ‌های یک فرآیند از پردازنده بهره‌مند می‌شوند و به محض مسدود شدن یک نخ، و یا پایان برش زمانی یک فرآیند، یا اتمام فرآیند، پردازنده به فرآیند بعدی تعلق می‌گیرد.

توجه: در این روش نخ ماهیت منطقی دارد و از دید کاربر فقط وجود دارد، در واقع از نظر سیستم عامل ماهیت فیزیکی ندارد، بنابراین نخ کاربر در این روش همانند یک تابع در فرآیند می‌باشد که از رجیستر و پشته مختص به خود نیز بهره‌مند نمی‌باشد.

توجه: مدل غیر کامپیوتری این روش نیز وجود دارد، مانند حالتی که درآمدهای دولت حاصل از منابع کشور، بین پدران خانواده‌ها تقسیم گردد و این پدران خانواده‌ها باشند که تصمیم بگیرند به هر عضو خانواده چه مقدار نقدینگی تعلق بگیرد. در این روش فقط پدران شماره حساب مختص به خود را دارند. اما اگر یکی از اعضای خانواده خطایی انجام دهد و محکوم گردد، آنگاه تمام اعضای خانواده برای مدتی از خدمات دولت محکوم می‌گردند، زیرا در این مدل، دولت فقط پدران خانواده را می‌شناسد و از اعضای خانواده اطلاعی ندارد. بنابراین حساب پدر خانواده برای مدتی مسدود می‌گردد.

توجه: عمل تعویض متن مابین نخ‌های یک فرآیند کاربر، کاملاً در سطح کاربر و با سربرار بسیار ناچیز (در حد فراخوانی نخ بعدی) و بدون تغییر حالت پردازنده به مد هسته پردازنده، توسط زمان‌بند چندنخی در سطح کاربر (برنامه‌های کاربر) انجام می‌گردد. اما عمل تعویض متن مابین فرآیندها (فرآیندهای کاربر یا فرآیندهای سیستم عامل) در سطح هسته سیستم عامل و در مد هسته پردازنده، توسط زمان‌بند پردازنده برای انتخاب یک فرآیند جدید بر اساس یک الگوریتم خاص انجام می‌گردد.

توجه: به دلیل آنکه نخ‌های یک فرآیند کاربر، تماماً در فضای کاربر مدیریت می‌شوند، اگر نخ موجود در یک فرآیند کاربر، یک فراخوان سیستمی مسدودکننده را اجرا نماید، هسته سیستم عامل نه تنها آن نخ، بلکه کل فرآیند کاربر را که شامل تمام نخ‌های دیگر می‌باشد، مسدود می‌کند، زیرا هسته سیستم عامل خبری از نخ‌های داخل فرآیند کاربر ندارد. در واقع در این روش هسته سیستم عامل نخ‌ها را همانند توابع داخل یک فرآیند می‌بیند، یعنی هسته سیستم عامل، یک فرآیند چند نخی سطح کاربر را، مانند یک فرآیند تک نخی اما دارای چند تابع مختلف می‌بیند! بنابراین در اینجا انتقال از حالت آماده نخ‌های همتای دیگر به حالت مسدود (منتظر) به دلیل اجرای فراخوان سیستمی مسدودکننده توسط یک نخ همتای دیگر را داریم، بنابراین گزینه دوم و سوم هم می‌تواند در این حالت درست باشد ولی نه در حالت کلی.

توجه: این راه‌کار، منجر به عدم امکان هم‌روندی (در سیستم‌های تک‌پردازنده‌ای) نخ‌های داخل یک فرآیند کاربر در حالت انسداد یک نخ داخل یک فرآیند کاربر می‌گردد.

البته اگر نخ‌های داخل یک فرآیند کاربر مسدود نگردد، امکان هم‌روندی میان نخ‌های داخل فرآیند کاربر برقرار است. مانند یک تیم فوتبال ۱۱ نفره که اگر بازیکنی مرتکب خطا گردد، از آن‌جا که داور فقط نام تیم را می‌شناسد و نه تک‌تک بازیکنان تیم را، آنگاه کل تیم را جریمه، اخراج و

مسدود می‌کند. اما اگر هیچ‌یک از بازیکنان تیم مرتکب خطایی نگردد، واضح است که هم‌روندی برقرار است.

توجه: فرض کنید یک کیک داریم که آن را به چهار قسمت مساوی تقسیم کرده‌ایم، همچنین فرض کنید خوردن هر بخش کیک یک ساعت زمان بخواهد، اگر یک نفر بخواهد تمام این کیک را بخورد، پس ۴ ساعت طول می‌کشد اگر ۴ نفر بخواهند تمام این کیک را بخورند و به هر نفر یک بخش کیک داده شود، آنگاه خوردن تمام کیک به‌طور موازی ۱ ساعت طول خواهد کشید. در روش سطح کاربر، یک فرآیند چند نخ‌ی کاربر نمی‌تواند از امتیازات چند پردازنده‌ای بهره‌برد، زیرا در روش سطح کاربر، هسته سیستم عامل در هر لحظه فقط یک پردازنده را در اختیار نخ‌های یک فرآیند کاربر قرار می‌دهد. حتی اگر چند پردازنده موجود باشد. یعنی در این روش خوردن کیک بخش‌بندی شده به صورت چند نفری امکان‌پذیر نیست. بنابراین در این روش امکان پردازش موازی نخ‌های داخل یک فرآیند کاربر وجود ندارد.

توجه: نرم‌افزارهای POSIX P-threads و Mach C-thread به عنوان یک بسته نرم‌افزاری، می‌توانند جهت مدیریت نخ‌ها و زمان‌بند چند نخ‌ی در سطح کاربر مورد استفاده قرار گیرند.

توجه: در این روش تخصیص منابع و زمان‌بندی پردازنده، بر روی فرآیندها انجام می‌شود. همچنین زمان‌بندی نخ‌های سطح کاربر، بر عهده زمان‌بند چندنخی سطح کاربر خواهد بود.

توجه: سیستم عامل سولاریس، مدل چند به یک را پیاده‌سازی می‌کند.

حذف نخی (Thread Cancellation)

حذف نخ‌ی به معنی **خاتمه‌دادن** یک نخ قبل از تکمیل شدن آن است. برای مثال اگر چند نخ بطور هم‌روند در حال جستجو در یک پایگاه داده باشند، در صورتی که یکی از نخ‌ها نتیجه را برگرداند، احتمالاً ادامه کار مابقی نخ‌ها کنسل می‌شود. حالت دیگر وقتی است که یک کاربر، دکمه‌ای را در یک مرورگر وب می‌فشارد که صفحه وب را از ادامه بارگذاری باز می‌دارد، اغلب یک صفحه وب با استفاده از چندین نخ بارگذاری می‌شود (هر تصویر توسط یک نخ مجزا بارگذاری می‌شود). وقتی که یک کاربر، دکمه stop را در مرورگر می‌فشارد، تمامی نخ‌های بارگذاری صفحه کنسل می‌شوند. نخ‌ی که باید کنسل شود اغلب **نخ هدف** (Target Thread) معرفی می‌شود. حذف (کنسل کردن) نخ هدف ممکن است در دو سناریوی مختلف رخ دهد:

۱- حذف ناهمگام (Asynchronous Cancellation)

یک نخ بلافاصله **نخ هدف** را خاتمه دهد، حذف نخ در این حالت به شکل ناایمن انجام می‌گردد، و بدون بررسی اشتراکات نخ حذف شده با نخ‌های دیگر.

بنابراین در این حالت، انتقال از حالت آماده نخ هدف به حالت خاتمه وجود دارد، پس گزینه اول هم می‌تواند درست باشد ولی نه در حالت کلی.

۲- حذف همراه تعویق (Asynchronous Cancellation)

نخ هدف به طور متناوب چک می‌شود که آیا می‌تواند خاتمه پیدا کند یا نه، و به آن اجازه داده می‌شود که بنا به درخواستش در صورت امکان به شکل امن، خاتمه یابد. حذف نخ در این حالت به شکل ایمن انجام می‌گردد، عموم سیستم‌عامل‌ها حذف همراه تعویق را پشتیبانی می‌کنند. به طور مثال سیستم عامل لینوکس حذف همراه تعویق را پشتیبانی می‌کند. بنابراین در این حالت، انتقال از حالت اجرای نخ هدف به خاتمه وجود دارد، ولی نه در حالت کلی.
