

فصل ۲:

فرآیند

در این فصل مفهوم فرایند و حالات فرایند بررسی می شوند. همچنین نخ نیز توضیح داده می شود.

فرایند و حالات آن

به برنامه در حال اجرا، فرآیند (process) می گویند. یک فرایند می تواند در یکی از حالت‌های زیر باشد:

۱- آماده (READY): فرایندی که وقتی به آن فرصت داده شود، برای اجرا آماده باشد.

۲- اجرا (Running): فرایندی که هم اکنون در حال اجر می باشد.

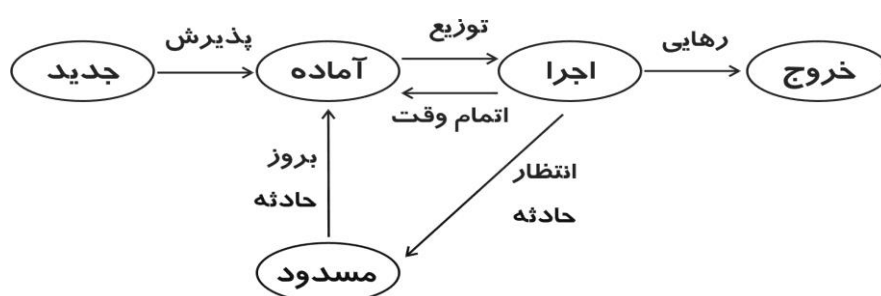
۳- مسدود (Blocked): فرایندی که تا بروز حادثه‌ای (مثل اتمام یک عمل I/O)، نمی تواند اجرا شود.

۴- جدید (New): فرایندی که هم اکنون گرفته شده است، اما هنوز جزء فرایندهای قابل اجرای سیستم عامل پذیرفته نشده باشد.

۵- خروج (Terminated): فرایندی که اجرای آن پایان یافته است و یا اجرای آن قطع شده و از مجموعه فرایندهای قابل اجرای سیستم عامل خارج شده است.

تذکر: به حالت مسدود، حالت بسته یا انتظار نیز می گویند.

در شکل زیر این حالات، نشان داده شده است:



تغییر حالات ممکن

۱- جدید به آماده

اگر سیستم عامل آمادگی گرفتن یک فرایند دیگر را داشته باشد، فرایند موجود در حالت جدید را به حالت آماده می برد.

۲- آماده به اجرا

سیستم عامل یکی از فرایندهای موجود در حالت آماده که وقت اجرای آن فرا رسیده است را انتخاب کرده و از حالت آماده به اجرا می‌برد. به این عمل توزیع (Dispatch) می‌گویند.

۳- اجرا به خروج

وقتی که فرایند جاری اعلام پایان کند، سیستم عامل آن را از حالت اجرا به خروج می‌برد.

۴- اجرا به مسدود

وقتی فرایندی چیزی را بخواهد که به خاطر آن باید منتظر بماند، سیستم عامل آن فرایند را از حالت اجرا به مسدود می‌برد. به این عمل بلوکه شدن می‌گویند. مثلاً فرایندی یک عمل I/O را شروع کند که قبل از تکمیل نیاز به بخش مشترکی از حافظه مجازی را داشته باشد که فوراً فراهم نباشد یا منتظر دریافت ورودی از یک فرایند دیگر باشد.

۵- مسدود به آماده

وقتی حادثه‌ای که فرایند منتظر آن بوده است رخ دهد، از حالت مسدود به آماده می‌رود. به این عمل wake up می‌گویند.

۶- اجرا به آماده

متداول ترین دلیل انتقال یک فرایند از حالت اجرا به آماده، اتمام زمان مجاز برای اجرای فرایند جاری در سیستم عامل‌های چند برنامه‌ای می‌باشد.

۷- آماده به خروج

در بعضی سیستم‌ها، یک پدر می‌تواند هر لحظه که بخواهد، فرایند فرزند خود را پایان دهد و یا با پایان یافتن فرایند پدر، ممکن است همه فرایندهای فرزند آن نیز پایان یابند.

۸- مسدود به خروج

مانند توضیحات آماده به خروج است.

۹- تهی به آماده

فرایند جدیدی را برای اجرای یک برنامه ایجاد می‌کند.

وجود وضعیت مسدود باعث افزایش بهره‌وری پردازنده می‌شود. چون وقتی فرایند در حال اجرا نیاز به I/O پیدا می‌کند، به حالت Blocking منتقل شده و فرایند آماده دیگری به قسمت اجرا منتقل می‌شود، تا در حد امکان CPU بیکار نماند.

دسته بندی فرایندها

بطور کلی فرایندها به دو دسته تقسیم می‌شوند:

۱- محدود به CPU (CPU Limited)

بیشتر زمان کامپیوتر صرف محاسبات CPU می‌شود.

۲- محدود به ورودی / خروجی (I/O Limited)

بیشتر زمان کامپیوتر صرف ورود داده‌ها و خروج اطلاعات می‌شود.

دلایل ایجاد یک فرایند جدید

۱- برقراری ارتباط محاوره‌ای

کاربر از طریق پایانه با سیستم ارتباط برقرار می‌کند.

۲- ارائه سرویس به وسیله سیستم عامل

سیستم عامل می‌تواند فرایندی را برای ارائه خدمتی از طرف برنامه کاربر ایجاد نماید، بدون اینکه کاربر ناچار به انتظار باشد.

۳- زایش توسط یک فرایند موجود

به منظور بهره‌گیری از توازی با تفکیک، برنامه کاربر می‌تواند ایجاد فرایندهای جدیدی را دیکته کند.

۴- کار دسته‌ای جدید

سیستم عامل با جریانی از کارهای دسته‌ای روبه‌رو می‌باشد. وقتی برای گرفتن یک کار جدید آماده است، دنباله بعدی از فرمانهای کنترل کار را می‌خواند.

عملیات ایجاد فرایند

۱- تخصیص یک شناسه یکتا به فرایند جدید ۲- تخصیص فضا برای فرایند

۳- مقدار دهی اولیه PCB ۴- برقراری پیوندهای لازم

۵- ایجاد و گسترش ساختمان داده‌های ممکن دیگر.

دلایل پایان یک فرایند

پایان طبیعی	فراخوانی یک سرویس سیستم عامل برای بیان تکمیل اجرایش
سقف زمانی	در فرایند محاوره ای، مقدار زمانی که از آخرین ورودی کاربر گذشته است.
گذشت زمان	انتظار زیادتر از حد برای بروز یک حادثه مشخص
نبود حافظه	نیاز به حافظه ای بیش از آنچه که سیستم می تواند فراهم کند.
تجاوز از حدود	مراجعه به محلهای غیر مجاز در حافظه
خطای حفاظت	تلاش برای دسترسی به منبعی که مجاز به استفاده از آن نیست.
خطای محاسباتی	مانند تقسیم بر صفر یا تلاش برای ذخیره عددی بزرگتر از ظرفیت سخت افزاری
خطای ورودی/خروجی	مانند پیدا نکردن یک فایل
دستورالعمل نامعتبر	تلاش برای اجرای دستورالعملی که وجود ندارد
دستورالعمل ممتاز	تلاش برای اجرای دستورالعملی که مخصوص سیستم عامل است
استفاده نامناسب از داده	داده با نوع نامناسب یا بدون مقدار اولیه
دخالت سیستم عامل	به دلایلی مانند بن بست
پایان یافتن پدر	با پایان یک فرایند، فرایندهای فرزند آن نیز پایان داده شوند
درخواست پدر	فرایند حق پایان دادن به هر یک از فرایندهای فرزند خود را دارد.

بلوک کنترل فرایند (PCB)

هر فرایند در سیستم بوسیله یک ساختاری به نام بلوک کنترل فرایند (Process Control Bolock) مشخص می‌شود. این بلوک مهمترین و محوریتترین ساختمان داده در سیستم عامل است که تمام اطلاعات مورد نیاز سیستم عامل در مورد یک فرایند را در بردارد.

PCB شامل موارد زیر است:

- ۱- حالت فرایند : یک فرایند می تواند در یکی از حالات جدید، آماده، اجرا، انتظار و غیره باشد.
- ۲- شماره برنامه : شامل آدرس دستور بعدی که باید اجرا شود.
- ۳- اطلاعات زمانبندی CPU : شامل اولویت فرایند و اشاره گر به صف زمانبندی.
- ۴- اطلاعات مدیریت حافظه : شامل مقدار ثباتهای پایه و حد، جدولهای صفحه یا قطعه.
- ۵- اطلاعات حسابرسی : میزان استفاده از پردازنده، محدودیتهای زمانی، شماره فرایند.
- ۶- اطلاعات وضعیت I/O : شامل لیست دستگاههای I/O تخصیص داده شده به فرایند
- ۷- ثباتهای CPU

تفاوت تعویض حالت با تعویض فرایند

تعویض حالت و تعویض فرایند (تعویض متن) دو مفهوم مجزا هستند. ممکن است تعویض حالت بدون تغییر حالت فرایندی که در حال اجراست، صورت گیرد. تعویض فرایند که متضمن تغییر حالت است، در مقایسه با تعویض حالت، به تلاش بیشتری نیاز دارد.

گام‌های که در یک تعویض فرایند کامل وجود دارد:

- ۱- ذخیره سازی متن پردازنده
- ۲- بهنگام سازی بلوکهای کنترل فرایندی که هم اکنون در حالت اجراست.
- ۳- انتقال بلوک کنترل فرایند مربوط به این فرایند به صف مناسب
- ۴- انتخاب فرایند دیگری برای اجرا
- ۵- بهنگام کردن بلوک کنترل فرایند مربوط به فرایند انتخاب شده
- ۶- بهنگام سازی ساختمان داده های مدیریت حافظه
- ۷- بارگذاری مجدد متن پردازنده

فرایند معلق

فرایند معلق، فرایندی است که فوراً آماده اجرا نیست. فرایند معلق توسط عاملی مانند سیستم عامل، خودش یا فرایند پدر، در حالت معلق قرار گرفته. تا وقتی عامل تعلیق فرمان ندهد، نمی توان فرایند را از حالت معلق خارج کرد.

دلایل معلق کردن یک فرایند

- ۱- **مبادله**: برای آوردن فرایندی که آماده اجراست، سیستم عامل نیاز به آزاد کردن حافظه کافی دارد.
- ۲- **ترتیب زمانی**: اجراء به طور دوره ای و به تعلیق رفتن آن هنگام انتظار برای اجرای بعدی
- ۳- **درخواست کاربر محاوره ای**: به منظور اشکالزدایی یا استفاده از منابع
- ۴- **دلایل دیگر سیستم عامل**: معلق کردن یک فرایند مضمون
- ۵- **درخواست فرایند پدر**: به طور نمونه فرایند A، فرایند B را تولید کرده تا عمل خواندن از یک فایل را انجام دهد و فرایند B در هنگام خواندن با خطایی مواجه شود و آنرا به فرایند A گزارش دهد، فرایند A در این حالت برای رسیدگی به علت خطا، فرایند B را به حالت معلق می برد.

نمودار تغییر حالت فرایند با حالات معلق

وقتی هیچ یک از فرایندهای موجود در حافظه اصلی در حالت آماده نیستند، سیستم عامل یکی از فرایندهای مسدود را از حافظه اصلی خارج و به صف فرایندهای معلق در روی دیسک می برد (مبادله) و فرایند دیگری را از صف معلق به حافظه اصلی می آورد یا درخواست فرایند جدیدی را می پذیرد و اجرا با این فرایند جدید ادامه می یابد.

وجود وضعیت معلق، موجب اجرای فرایند جدید حتی در صورت پر بودن حافظه اصلی می شود.

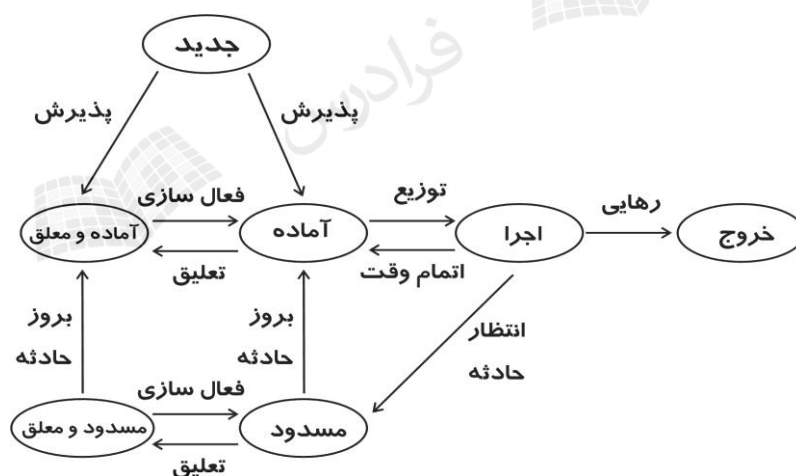
بنابراین حالت معلق را به حالت‌های گفته شده، اضافه می کنیم:

۱- مسدود و معلق (Suspend-Wait)

فرایند مورد نظر در حافظه ثانوی و منتظر حادثه ای است.

۲- آماده و معلق (Suspend-Ready)

فرایند مورد نظر در حافظه ثانوی و به محض لود شدن در حافظه اصلی آماده اجراست.



حالات ممکن در این نمودار

۱- مسدود به مسدود و معلق

وقتی هیچ فرایند آماده ای وجود نداشته باشد، فرایند مسدودی به خارج برده می شود تا جا برای فرایند دیگری که مسدود نیست فراهم شود. این تغییر حالت حتی در مواردی که فرایندهای آماده هم وجود دارد می تواند جهت حفظ کارایی انجام شود.

۲- مسدود و معلق به آماده و معلق

وقتی حادثه ای که یک فرایند مسدود و معلق منتظر آن بوده است رخ دهد به حالت آماده و معلق می‌رود.

۳- آماده و معلق به آماده

وقتی که هیچ فرایند آماده ای در حافظه اصلی نباشد، سیستم عامل یک فرایند آماده و معلق را به حالت آماده می آورد. همچنین ممکن است فرایند موجود در حالت آماده و معلق دارای اولویت بیشتری نسبت به همه فرایندهای آماده باشد که در این حالت فرایند به حالت آماده، آورده می شود.

۴- آماده به آماده و معلق

به طور معمول سیستم عامل ترجیح می دهد یک فرایند مسدود را به جای فرایند آماده، به حال تعلیق در آورد. ولی در صورتی که راهی برای خالی کردن حافظه اصلی نباشد، یک فرایند آماده را به تعلیق در می آورد. سرانجام ممکن است سیستم عامل یک فرایند آماده ولی با اولویت کم را به جای فرایند مسدودی که اولویتش بیشتر است و گمان می کند بزودی آماده می شود به حالت معلق می برد.

۵- مسدود و معلق به مسدود

اگر اولویت فرایندی که در صف مسدود و معلق است از اولویت همه فرایندهای موجود در صف آماده و معلق بیشتر باشد و سیستم عامل گمان کند که حادثه ای که این فرایند مسدود، منتظر آن بوده است به زودی رخ دهد، آن را به حالت مسدود می آورد. البته باید مقداری از حافظه اصلی خالی باشد تا آوردن یک فرایند مسدود به حافظه نسبت به یک فرایند آماده معقول به نظر برسد.

۶- اجرا به آماده و معلق

معمولا با پایان زمان منظور شده برای فرایند جاری، فرایند به حالت آماده منتقل می شود. در اینحال اگر این فرایند به خاطر فرایند با اولویت بیشتری قبضه شود، سیستم عامل می تواند فرایند جاری را مستقیما به صف آماده و معلق منتقل کند تا بخشی از حافظه اصلی آزاد شود.

۷- مختلف به خروج

ممکن است یک فرایند از هر حالتی به حالت خروج منتقل شود.

انواع زمانبندها

کلید چند برنامه‌ی زمانبندی است. زمانبندی بر روی کارایی سیستم اثر می‌گذارد، زیرا مشخص می‌کند کدام فرایندها منتظر مانده و کدام فرایندها به جلو بروند. انواع زمانبندی برای پردازنده عبارت است از:

۱- زمانبند بلند مدت (Long Term Scheduler)

تصمیم‌گیری در مورد افزودن به مجموعه فرایندها برای اجرا.

۲- زمانبند میان مدت (Middle Term Scheduler)

تصمیم‌گیری در مورد افزودن به تعداد فرایندهایی که بخشی یا تمام آنها در حافظه اصلی است.

۳- زمانبند کوتاه مدت (Short Term Scheduler)

تصمیم‌گیری در مورد اینکه کدام یک از فرایندهای موجود در حافظه اصلی، برای اجرا توسط پردازنده انتخاب شود.

۴- زمانبندی ورودی/خروجی

تصمیم‌گیری می‌گیرد که کدام درخواست I/O فرایندها به وسیله یک دستگاه I/O موجود انجام بگیرد.

✓ وظیفه فعال‌سازی و تعلیق فرایندها بر عهده زمانبند میان مدت (Medium-term scheduler) می‌باشد.

✓ زمانبند میان مدت، فرایندی را از حافظه اصلی حذف و به حافظه جانبی می‌برد. این فرایند بعداً می‌تواند به حافظه اصلی لود شود. این الگو را مبادله (swapping) می‌گویند.

✓ ایده اصلی زمانبندی میان مدت، این است که می‌تواند فرایندی را از حافظه حذف کند و درجه چند برنامه‌ی را کاهش دهد.

✓ زمانبند بلند مدت، ترکیب خوبی از فرایندهای I/O limited و CPU Limited، انتخاب می‌کند.

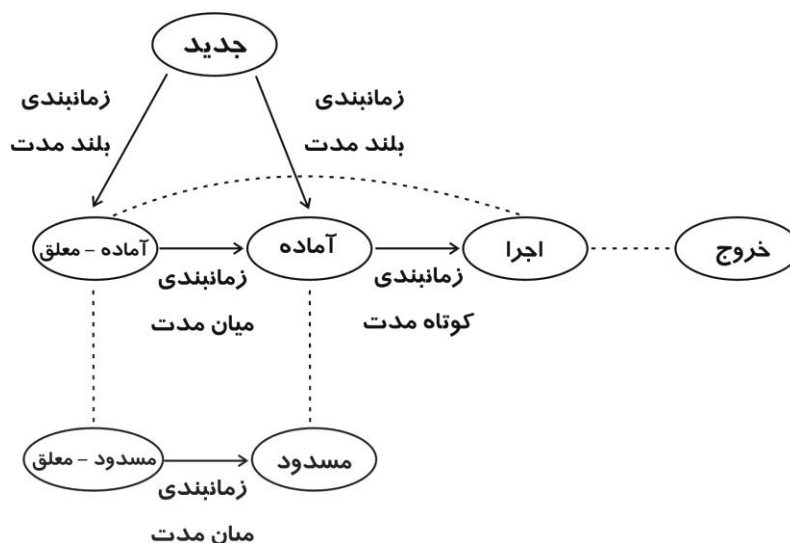
✓ نام دیگر زمانبند بلند مدت، زمانبند کار است.

✓ نام دیگر زمانبند کوتاه مدت، زمانبند پردازنده است.

✓ زمانبند بلند مدت نسبتاً دفعات کمی اجرا می‌شود، زمانبند میان مدت نسبتاً دفعات بیشتری به اجرا در می‌آید و زمانبند کوتاه مدت، بیشترین دفعات اجرا را دارد.

تذکر: اکثر سیستم‌های اشتراک زمانی فاقد زمانبند بلند مدت می‌باشند.

نمودار تغییر حالت فرایند همراه با زمانبندی



مثال

اگر ۱۰ میلی ثانیه طول بکشد تا فرایندی انتخاب گردد که ۱۰۰ میلی ثانیه اجرا شود، آنگاه چند درصد از زمان CPU، صرف زمانبندی کار می شود (به هدر می رود)؟

$$\frac{10}{10 + 100} \times 100 = 9\%$$



توزیع کننده (Dispatcher)

توزیع کننده، پیمانه ای است که کنترل را به پردازنده ای می دهد که توسط زمانبند کوتاه مدت انتخاب شده است. این عمل شامل موارد زیر است:

۱- تعویض بستر (Context Switch)

۲- تغییر به حالت کاربر

۳- پرش به محل مناسبی در برنامه کاربر و آغاز مجدد آن برنامه.

نخ (Thread)

تا به حال، مفهوم فرایند را به صورت واحدی معرفی کرده ایم که دارای دو خصوصیت تملک منبع و توزیع وقت پردازنده است. این دو خصوصیت می توانند مستقلاً توسط سیستم عامل رسیدگی شوند. برای تمایز این دو خصوصیت، به توزیع وقت پردازنده، نخ و به تملک منبع، فرایند می گویند.

مهم ترین مزیت استفاده از سیستم های چند نخ این است که در فرایند تک نخ، هر گاه فراخوان سیستمی مسدود کننده ای اجرا شود، یا یک نقص صفحه رخ دهد، کل فرایند مسدود می شود. به عنوان مثال یک برنامه صفحه گسترده را در نظر بگیرید و فرض کنید کاربری می خواهد به طور دائمی و تعاملی مقادیر را تغییر دهد. می دانیم که در برنامه صفحه گسترده، وابستگی تابعی بین سلول های مختلف باید حفظ شود. بنابراین اگر سلولی تغییر کند، محاسبات زیادی انجام می گیرد. اگر فقط یک نخ وجود داشته باشد، وقتی برنامه منتظر دریافت ورودی است، محاسبات نمی تواند ادامه یابد، چون فراخوان سیستمی دریافت ورودی، کل فرایند را مسدود کرده است. راه حل این است که از یک نخ برای خواندن ورودی کاربر و از نخ دیگری برای بهنگام سازی صفحه گسترده استفاده شود.

مثال

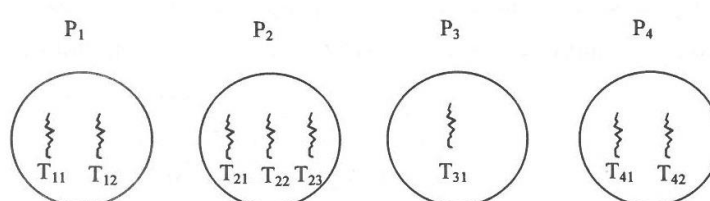
بسیاری از صفحات وب، شامل چند تصویر کوچک هستند که مرورگر باید یک اتصال جداگانه را برای دریافت هر تصویر برقرار کند که زمان زیادی لازم دارد. اگر در مرورگر از روش چند نخ استفاده شود، هر کدام از نخ ها می تواند به طور همزمان با نخ های دیگر تصویر مربوط به خود را درخواست کند. به عنوان مثالی دیگر از کاربرد نخ ها، جهت حفاظت در مقابل قطع برق می توان یک کلمه پرداز را طوری طراحی کرد که هر یک دقیقه بافر RAM را روی دیسک بنویسد. یک نخ می تواند تنها برای گرفتن پشتیبان دوره ای ایجاد شود و خودش را مستقیماً با سیستم عامل زمانبندی کند.

نخها و فرایندها می توانند ۴ حالت را بوجود آورند:

- ۱- یک فرایند- یک نخ
- ۲- یک فرایند - چند نخ
- ۳- چند فرایند- یک نخ در هر فرایند
- ۴- چند فرایند- چند نخ در هر فرایند

مثال

شکل زیر سیستمی با چهار فرایند را نشان می دهد که فرآیند اول شامل دو نخ T_{11} و T_{12} می باشد. فرایند دوم شامل سه نخ است و ...



حالات اصلی نخ عبارت است از: اجرا، آماده و مسدود. (نخ حالت معلق ندارد)

تمام نخ‌های یک فرایند، در حالت و منابع آن فرایند شریک هستند.

اگر نخ یک فرایند در حال اجرا باشد و آن فرایند به حالت خروج برود، نخ نمی‌تواند به اجرا ادامه دهد.

تعویض متن میان دو نخ متعلق به دو فرایند جداگانه، مثل این است که تعویض متن فرایند رخ داده است. یعنی عمل سوئیچینگ مابین دو نخ متعلق به دو فرایند جداگانه از نوع تعویض متن فرایندی است.

نخ در برنامه نویسی، بخشی از یک فرایند یا برنامه بزرگتر می‌باشد.

ساختار نخ گاهی در محیط تک پردازنده‌ای، برای ساده کردن ساختار برنامه‌ای که منطقاً اعمال متعددی را انجام می‌دهد، نیز مفید است.

با تقسیم یک کار به چند نخ، برنامه‌ساز می‌تواند کنترل زیادی روی مولفه‌ای بودن آن کاربرد و تنظیم وقت حوادث مربوط به آن داشته باشد.

پیاده سازی نخ

سه روش برای پیاده سازی و مدیریت نخ وجود دارد:

۱- نخ‌های سطح کاربر

۲- نخ‌های سطح هسته

۳- روش‌های ترکیبی

نخ‌های سطح کاربر

در این روش تمام عملیات راهبری در فضای آدرس کاربر انجام می‌شود. در این روش هزینه ایجاد نخ با هزینه تخصیص حافظه برای برپا سازی پشته (stack) نخ تعیین می‌شود. تعویض متن نخ اغلب با تعداد کمی از دستورات العمل‌ها انجام می‌شود و فقط کافی است محتوای ثبات‌های پردازنده برای نخ فعلی، ذخیره شود و سپس مقادیر قبلاً ذخیره شده مربوط به نخ‌ی که به آن سوئیچ می‌شود، بارگذاری شوند. در تعویض متن نخ‌ها نیازی به فلاش کردن TLB، تغییر نگاشت‌های حافظه و حسابداری پردازنده نمی‌باشد.

مزایای کتابخانه نخ سطح کاربر

۱- ایجاد و حذف سریع نخ‌ها

۲- همگام سازی سریع نخ‌ها

۳- تعویض متن کم هزینه و سریع

نقاط ضعف نخ‌های سطح کاربر

- ۱- اگر یک نقص صفحه (page fault) برای یک نخ رخ دهد، کل فرایند و در نتیجه همه نخ‌های درون آن به اشتباه مسدود می‌شود.
- ۲- اگر نخ‌ی یک فراخوان سیستمی بلوکه کننده را صدا بزند، کل فرایند و همه نخ‌های درون آن نیز به اشتباه مسدود می‌شوند.
- ۳- چون سیستم عامل از وجود نخ‌ها آگاه نمی‌باشد، نخ‌ها را بین چندین پردازنده به خوبی پخش و زمان بندی نمی‌کند.

نخ‌های سطح هسته

اگر نخ‌ها در هسته سیستم عامل پیاده سازی شوند، مشکلات روش قبل رخ نمی‌دهند. اما چون هر عملیات نخ باید توسط هسته انجام شود، نیاز به یک فراخوانی سیستمی دارد و هزینه بالا می‌رود. تعویض متن نخ در این روش ممکن است به اندازه تعویض متن فرایند پر هزینه باشد.

تذکر: مزایا و معایب نخ‌های سطح کاربر و نخ‌های سطح هسته، بر عکس یکدیگرند.

روش ترکیبی

برای غلبه بر مشکلات دو روش قبل، نخ‌های سطح کاربر و سطح هسته را را تلفیق می‌کنیم. برای این کار فرایندهای سبک وزن (LWP) ایجاد می‌کنیم. LWP دورن متن یک فرایند اجرا شده و به ازای هر فرایند ممکن است چند LWP وجود داشته باشد. سیستم علاوه بر داشتن LWP ها، یک بسته نخ سطح کاربر برای عملیات ایجاد و حذف نخ‌ها به برنامه‌های کاربردی ارائه می‌دهد. بسته نخ کاملاً در فضای کاربر پیاده سازی می‌شود و تمام عملیات روی نخ‌ها بدون دخالت هسته انجام می‌شود. برنامه‌های کاربردی چند نخ‌ی با ایجاد نخ‌ها و سپس انتساب هر نخ به یک LWP ساخته می‌شوند.

در این روش اگر نخ‌ی فراخوان سیستمی مسدود کننده‌ای را اجرا کند، اجرا از مد کاربر به مد هسته تغییر می‌کند ولی همچنان در متن LWP جاری ادامه می‌یابد. در جایی که LWP جاری نتواند ادامه دهد، متن به LWP دیگری تعویض می‌شود که باعث یک تعویض متن برای بازگشت به مد کاربر نیز می‌شود.

 در روش ترکیبی، فراخوانی‌های سیستمی از نوع مسدود با تامین هم‌روندی حمایت می‌شوند.

تغییر متن (context switch) در نخ‌های یک فرایند، اشاره‌گر پشته (SP) را تغییر می‌دهد ولی ثبات‌ها و جداول مدیریت حافظه را تغییر نمی‌دهد و ارتباطی با TLB ندارد.

TLB یک سخت‌افزار جستجوی سریع، کوچک و پنهان است. در صفحه‌بندی تعداد اندکی از ورودیهای جدول صفحه در TLB قرار می‌گیرد.

(TLB : Translation Lookaside Buffers)