

哈工大操作系统-L10用户级线程

哈工大操作系统-L10用户级线程

1.线程的引出

1.1进程为何要分出线程

1.2 线程有什么价值

2.如何实现用户级线程的切换

3.引出内核级线程

本课的核心是：用户级线程(user threads)之间的切换，主要是指令如何切换。而进程之间的切换=线程切换+映射表切换。

1.线程的引出

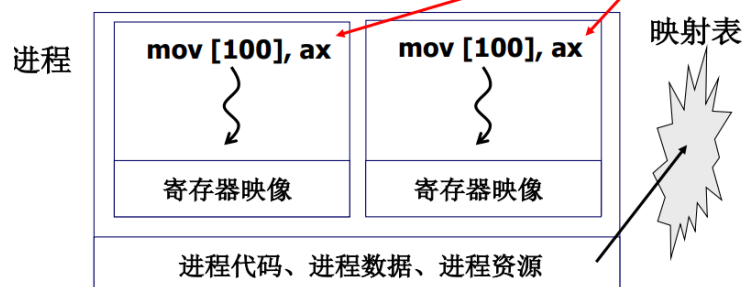
1.1进程为何要分出线程

- 进程=资源+指令执行序列
- 我们可以把资源和指令执行序列分开
- 进程中的多个指令序列分开，他们共享同样的资源却是不同的指令，我们把这些不同的指令序列叫线程

是否可以资源不动而切换指令序列？

■ 进程 = 资源 + 指令执行序列

- 将资源和指令执行分开
- 一个资源 + 多个指令执行序列



如果两段指令需要访问的都是同一个资源，那么再进行资源的切换就显得多余了。因此可以把资源和指令执行分开，如果一个进程中有多指令执行序列是访问共同的资源，则可以使用线程

■ 线程：保留了并发的优点，避免了进程切换代价

■ 实质就是映射表不变而PC指针变



1.2 线程有什么价值

比如一个浏览器显示网页，那么需要把这个网页的数据下载下来然后在屏幕上显示出来。

- 那么下载下来的东西如果分散放就太离谱了，所以把一个网页的东西放在一个地方。
- 如果我们只设计一个进程，先下载，再读取，再显示，那么可能效果就不太好
 - 可能网页下载文字快而下载图片慢
- 如果我们设计一个显示的线程，设计一个下载文字的线程，设计一个下载图片的线程
 - 先下载文字，再使用显示的线程，再下载图片，再使用显示的线程

- 那么用户就可以先看到文字，在看文字的同时等待图片的下载，对于用户而言，等待时间感觉就少了

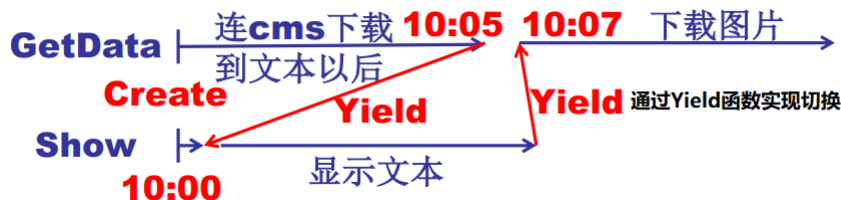
2.如何实现用户级线程的切换

因为是用户级线程，所以我们暂时不关心内核是如何实现切换的，那部分会在内核级线程中讲。

开始实现这个浏览器...

```
void WebExplorer()
{
    char URL[] = "http://cms.hit.edu.cn";
    char buffer[1000]; 共享的缓冲区
    pthread_create(..., GetData, URL, buffer); 下载的线程
    pthread_create(..., Show, buffer); 显示的线程
}

void GetData(char *URL, char *p){...};
void Show(char *p){...};
```



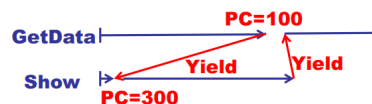
Create? Yield?

■ 核心是Yield...

- 能切换了就知道切换时需要是个什么样子
- Create就是要制造出第一次切换时应该的样子

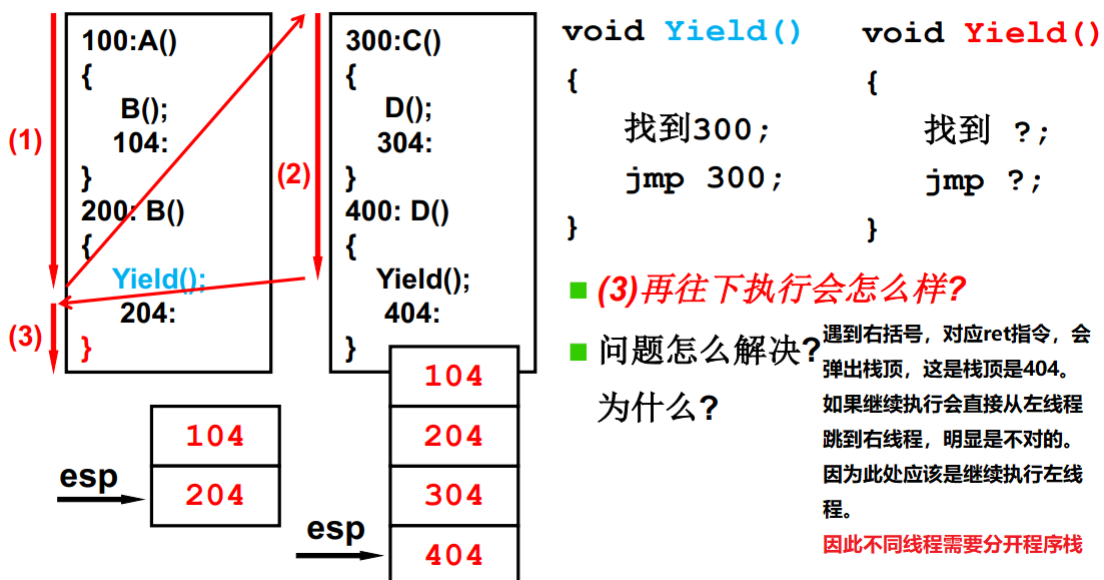
样子弄明白了，剩下的就是写程序实现这个样子了...

■ 仔细看Yield，就是100跳到300



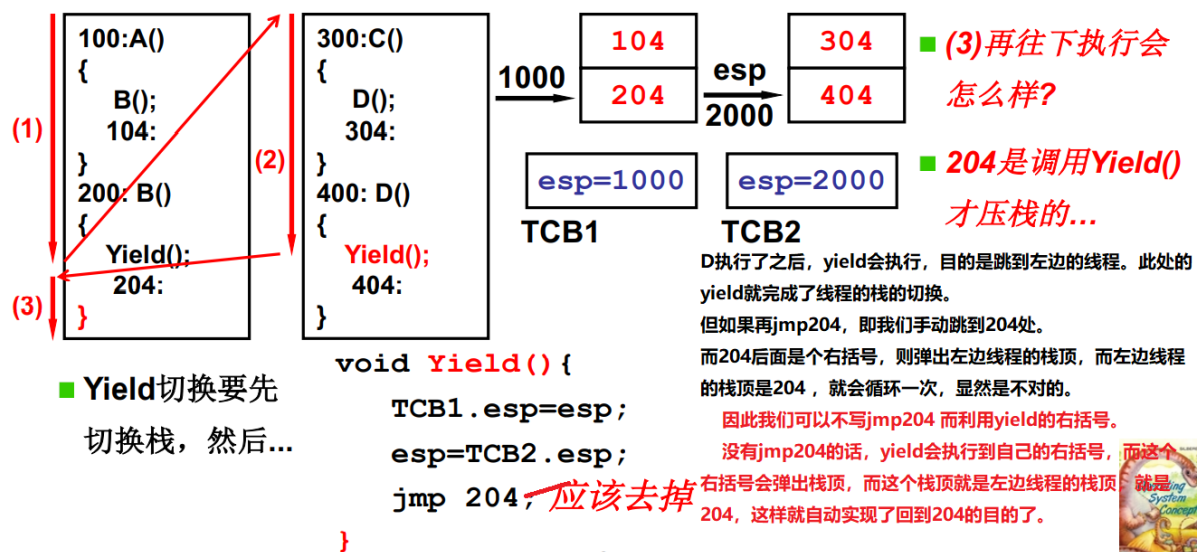
所以实现用户级线程切换的核心在，如何实现yield和creat函数。

两个执行序列与一个栈...



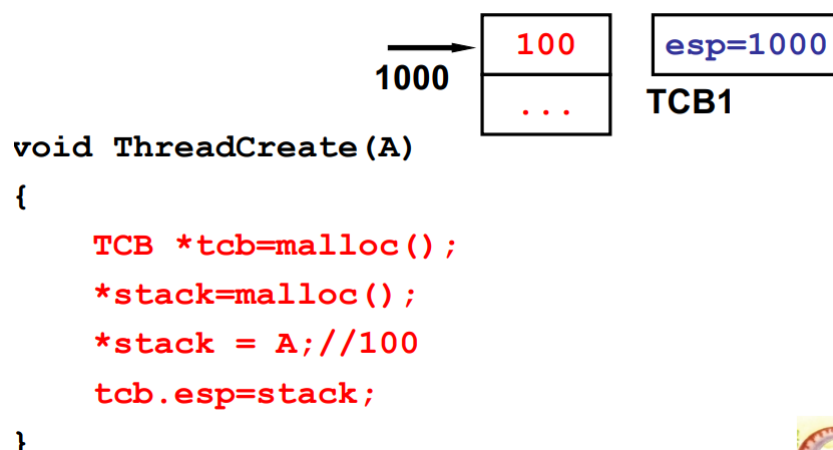
从上图可知, 不同的线程需要各自的函数栈。为了存放各自的栈, 就又引出了一个数据结构TCB(thread control block)。

从一个栈到两个栈...



上图讲了yield大概的实现思路。

- ThreadCreate的核心就是用程序做出这三样东西



而creat就只需要为每个线程申请各自的TCB以及线程栈即可。

3.引出内核级线程

用户级线程不涉及与内核交互，就纯粹是在用户态反复横跳creat yield就行。内核不会感知到用户级线程的存在。

如果线程只停留在用户态，那当一个进程的用户线程要请求IO或其他内核操作时，就会阻塞，而当一个进程的用户线程阻塞时，会导致跳转到另一个进程，那么此进程的其他用户线程就都阻塞，有时这不是我们想要的效果。所以需要引出内核级线程。