

## 4.2

类似于内核线程之间的切换。简单的说，涉及将一个用户线程从其轻量级进程（LWP）中移除，并用另一个线程替换它。具体的说，1. 保存当前线程的上下文。2. 选择下一个要执行的线程。3. 恢复下一个线程的上下文。4. 切换栈指针和寄存器。5. 切换控制权。

### 补充 1

#### 1. 线程管理层次不同：

用户级线程：线程的创建、调度和管理完全由用户态的线程库负责。

内核级线程：线程的管理由操作系统内核直接处理，内核知道所有线程的存在，并负责它们的调度和管理。

#### 2. 上下文切换的开销不同：

用户级线程：上下文切换在用户态完成，不涉及内核态切换，切换速度较快，开销较小。

内核级线程：上下文切换需要在用户态和内核态之间进行，涉及系统调用，开销较大。

#### 何时选择哪种线程更好？

用户级线程适合在单核环境中或当线程切换需要非常高效时使用。

内核级线程适合在多核环境或需要并行执行多个线程的场景下使用。

### 补充 2

线程创建：栈空间、线程控制块、线程局部存储

进程创建：内存空间、进程控制块、文件描述符表、资源分配

主要区别：进程拥有独立的内存空间，因此创建进程时需要分配大量的内存资源；而线程是共享进程的地址空间，因此线程创建消耗的内存资源较少。创建进程的开销比创建线程大，因为进程需要独立的资源和地址空间，而线程共享进程的资源，创建速度更快，开销更小。