

第四章课后练习

4.1 Provide three programming examples in which multithreading provides better performance than a single-threaded solution.

- ① 服务器进程每次收到一个请求就开启一个新线程来处理对应请求，相比单线程，可以并行处理任务，具有更好的并发功能，不容易阻塞。
- ② 带GUI的程序中，多线程可以支持同时进行复杂运算和维持GUI的反应速度，而单线程的话进行复杂计算时可能会阻塞使得GUI不灵敏。
- ③ 在多核处理器的系统中，进行可并行化程度较高，相对比较独立的大量数据计算时，采用多线程可以更好地利用多核CPU，显著提高运算速度。

4.4 What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other?

difference:

用户线程位于内核之上，它的管理无需内核支持，而内核线程则直接由操作系统来进行支持与管理

circumstance:

- ① 线程切换时，用户级线程的切换无需经过内核，而内核级线程的切换则需要转为内核态，完成切换后再转为用户态，因此内核级线程的切换比用户级线程的切换要更昂贵。从这方面来看，用户级线程要比内核级线程好
- ② 内核级线程可以更好地利用多核CPU，而用户级线程在这方面做的没有内核级线程好

4.10 Which of the following components of program state are shared across threads in a multithreaded process? a. Register values b. Heap memory c. Global variables d. Stack memory

answer: c. Global variables

4.17

a. How many unique processes are created?

b. How many unique threads are created?

a. 加上初始的进程，总共有六个进程

b. 总共有10个线程

4.19 The program shown in Figure 4.22 uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

LINE C: CHILD: value = 5

LINE P: PARENT: value = 0