



22920192204097

厦 門 大 學

XIAMEN UNIVERSITY

ADD:FUJIAN XIAMEN

CABLE:0633 P.C:361005

4.1

解: 是. 因为从一个进程切换到另一个进程必须保存更多的状态信息, 所以同一个进程中的两个线程切换的开销比不同进程中的两个线程切换的开销少.

4.2

解: 因为对于用户级线程来说, 一个进程的线程结构对操作系统是不可见的, 而操作系统的调度是以进程为单位的.

4.4

解: 阻塞会花费相当多的时间在等待 I/O 上. 在多线程程序中, 一个内核级线程可能会产生引发阻塞的系统调用, 其他内核级线程可以继续运行. 在单处理上, 如果一个进程因为系统调用而阻塞, 那么它可以继续运行其他线程.

4.5

解: 不会. 当一个进程退出时, 它会清除相应的进程结构、内存空间等, 包括线程. 所以其正在运行的线程不会继续运行.

4.7

解: a. 该函数的作用是统计链表中正数的个数.

b. 不会. 因为线程 A 执行操作时链表中全是负数, 所以不会更新全局变量 global_positives. 只有线程 B 执行的操作会更新该变量. 所以两个线程不会互相干扰.



扫描全能王 创建



厦 門 大 學

XIAMEN UNIVERSITY

ADD: FUJIAN XIAMEN

CABLE: 0633 P. C: 361005

4.8

解: 线程 A 中的寄存器变量 r 保存了全局变量 global-positives 的值, 线程 B 修改 global-positives 的值之后, 可能会被线程 A 修改为寄存器 r 中的值, 使得 global-positives 的值产生错误.

例如以下的执行顺序:

A 执行:

```
list p;  
register int r;  
r = global-positives;
```

B 执行:

```
++ global-positives;
```

A 执行:

```
for (p = l; p; p = p->next)  
    if (p->val > 0.0) ++r;  
global-positives = r;
```

这样 A 线程就把 B 线程修改的 global-positives 值又改回去了, 就会产生错误.



扫描全能王 创建