

操作系统原理第四章作业

姓名：马福泉 学号：23336179 截止日期：2025年4月9日

完成日期：2025年4月7日

Question 1：描述线程库进行用户级线程上下文切换的过程所采取的措施。

Answer 1：

- (1) 触发切换：主动调用、信号中断。
- (2) 保存当前线程的上下文：保存当前线程的寄存器状态，栈指针，程序计数器等。
- (3) 调度器选择新线程：根据策略从就绪队列选取一个线程，切换更新到新的栈指针和栈空间。
- (4) 恢复新线程的上下文：从新线程的 TCB 中加载恢复寄存器状态和程序计数器等。
- (5) 切换执行流：跳转到新线程的 EIP 地址，继续执行。

Question 2：在什么环境中，采用多内核线程的多线程方法比单处理器系统的单线程提供更好的性能？

Answer 2：

- (1) 分析：当内核线程发生页面错误时，采用多内核线程的多线程方法可以切换到另一个内核线程，以一种有用的方式使用交错时间。而另一方面，单处理器系统的单线程将无法执行有用的工作。
- (2) 结论：因此，在程序可能出现频繁的页面错误或必须等待其他系统事件的情况下，采用多内核线程的多线程方法比单处理器系统的单线程提供更好的性能。
- (3) 举例：计算密集型任务，高并发 I/O 操作，多任务重叠执行等等。

Question 3：在多线程进程中，下列哪些程序状态组成被共享？

- a. 寄存器值
- b. 堆内存
- c. 全局变量
- d. 栈内存

Answer 3：

- (a) 寄存器值（不共享）：寄存器是线程私有的，每个线程有自己的寄存器状态。当线程切换时，操作系统会保存当前线程的寄存器状态，并恢复下一个线程的寄存器状态。
- (b) 堆内存（共享）：堆内存是进程级别的资源，由操作系统分配给整个进程使用。所有线程都可以访问和操作堆内存中的数据。
- (c) 全局变量（共享）：全局变量存储在全局数据区，所有线程共享同一份全局变量。
- (d) 栈内存（不共享）：每个线程都有自己独立的栈内存，用于存储局部变量和函数调用的上下文信息。栈是线程私有的，不同线程的栈是独立的。

Question 4: 考虑多处理器系统和采用多对多线程模式编写的多线程程序，使程序中用户级线程数比系统中处理器数多。讨论下列情形的性能影响：

- a. 分配给程序的内核线程数比处理器数少。
- b. 分配给程序的内核线程数与处理器数相等。
- c. 分配给程序的内核线程数比处理器数多，但少于用户线程数。

Answer 4:

- (a) 分配给程序的内核线程数比处理器数少
 - ① 部分处理器会处于空闲状态，导致处理器资源浪费。
 - ② 由于内核线程数量有限，用户线程需要在有限的内核线程之间进行调度和切换，导致并发性下降。
 - ③ 较少的内核线程意味着较少的调度竞争，减少了内核级上下文切换的开销。
- (b) 分配给程序的内核线程数与处理器数相等
 - ① 每个处理器可以分配到一个内核线程，理论上可以充分利用每个处理器的计算能力。
 - ② 由于用户线程数多于内核线程数，用户线程需要在内核线程之间进行调度和切换，如果内核线程因阻塞而无法执行，对应的处理器将闲置。
 - ③ 频繁的线程切换也会增加上下文切换的开销，从而影响性能。
- (c) 分配给程序的内核线程数比处理器数多，但少于用户线程数

- ① 内核线程数量多于处理器数，可以更好地利用多处理器系统的资源。
- ② 当某个内核线程被阻塞时，调度器可以快速切换到另一个准备就绪的内核线程，从而减少处理器的闲置时间，提高系统的利用率。
- ③ 由于内核线程数量仍然少于用户线程数量，线程切换仍然是不可避免的。