

OS 第二次作业

编写一个 C/S 架构的分布式程序, Server 接收 Client 发来的请求, 执行一个计算 $F(X)$ 并给 Client 返回结果; 分别用进程与线程作为服务器 Server 实现, 并比较服务器的开销, 可以在一台机器上模拟。

定义 $cal(x) = x^2 - x + 1$, 分别用进程与线程作为服务器 Server 实现计算和回复请求, 代码见 Client.java, jcServer.java(进程实现), Server.java(线程实现)。

jcServer.java 运行结果:

```
Result from server: 31
Time taken: 22562800 nanoseconds
Result from server: 31
Time taken: 21358600 nanoseconds
Result from server: 31
Time taken: 22403900 nanoseconds
Result from server: 31
Time taken: 22340700 nanoseconds
Result from server: 31
Time taken: 21790700 nanoseconds
```

Server.java 运行结果:

```
Result from server: 31
Time taken: 21966600 nanoseconds
Result from server: 31
Time taken: 20097600 nanoseconds
Result from server: 31
Time taken: 19977700 nanoseconds
Result from server: 31
Time taken: 19775800 nanoseconds
Result from server: 31
Time taken: 20061500 nanoseconds
```

线程实现的运行结果中, 除第一个连入的Client与进程相比开销接近, 其余均有明显差距(当只有一个线程创建时, 开销差距不明显)。比较二者开销, 线程实现开销更小。

5.1 为什么区分CPU密集型程序和I/O密集型程序对调度程序是重要的?

I/O密集型程序具有大量短CPU执行, 需要大量时间用于I/O等待而CPU密集型只有少量长CPU执行, 不阻碍IO操作。通过区分他们, 进行合理调度, 可以提高CPU的资源利用率。给I/O密集型程序更高的优先级, 就可以使得CPU操作后台同时运行I/O等待操作。

5.4 本章讨论针对多个内核数据结构的可能竞争条件。大多数调度算法采用一个运行队列(run queue)，用于维护可在处理器上运行的进程。对多核系统，有两个常用选择:1) 每个处理核都有各自的运行队列，或2)所有处理核共享一个运行队列。这些方法的优点和缺点是什么？

1) 优点：不会出现对单一运行队列的争夺情况，不易冲突；缺点：管理不便。

2) 优点：管理方便；缺点：需要进行同步来避免对单一运行队列的争夺，容易冲突。

5.5 假设采用指数平均公式来预测下个 CPU 执行的长度。当采用如下参数数值时，该算法的含义是什么：

a. $\alpha = 0$ 和 $\tau_0 = 100\text{ms}$;

b. $\alpha = 0.99$ 和 $\tau_0 = 10\text{ms}$ 。

a. 代入 $\alpha = 0$ 得 $\tau_{n+1} = \tau_n = \dots = \tau_0 = 100\text{ms}$ ，算法含义为总是预测 CPU 执行长度为固定的 100ms。

b. 代入 $\alpha = 0.99$ 得 $\tau_{n+1} = 0.99\tau_n + 0.01t_n$ ， τ_n 对于 τ_{n+1} 的影响几乎可以忽略不计，算法含义为预测下次 CPU 执行长度为上次实际执行长度。

课堂小测验：根据进程状态图分析并回答分时系统与实时系统实现上的区别。

分时系统：进程调度算法为轮转算法，将CPU运行时间划分为多个时间片，每个进程在CPU上运行一个时间片，然后切换给下一个进程。

ready队列为一个FIFO队列，scheduler dispatch为取队头，分配给cpu上运行一个时间片，若进程只需要不到一个时间片的运行时间，则exit释放，若一个时间片后进程没有运行完，则对应图中interrupt中断，放入ready队列队尾继续等待。多个进程频繁切换，实现了多个进程同时处理的效果。

实时系统：进程调度算法为优先级调度，每个进程设有一个优先级，分配时将优先级最高的进程分配给CPU。

其实现上，图中的 ready 队列为一个优先队列，队顶为优先级最高进程。scheduler dispatch 操作对应取优先级最高进程给其分配 CPU。提高了CPU的利用率。