Java 核心技术(进阶)

第五章 Java多线程和并发编程

第一节 多进程和多线程简介

华东师范大学 陈良育

多进程概念(1)

- 当前的操作系统都是多任务OS
- 每个独立执行的任务就是一个进程
- OS将时间划分为多个时间片（时间很短）
- 每个时间片内将CPU分配给某一个任务，时间片结束，CPU将自动回收，再分配给另外任务。从外部看，所有任务是同时在执行。但是在CPU上，任务是按照串行依次运行（单核CPU）。如果是多核，多个进程任务可以并行。但是单个核上，多进程只能串行执行。

多进程概念(2)

- 多进程的优点
  - 可以同时运行多个任务
  - 程序因IO堵塞时，可以释放CPU，让CPU为其他程序服务
  - 当系统有多个CPU时，可以为多个程序同时服务
    - 我们的CPU不再提高频率，而是提高核数
    - 2005年Herb Sutter的文章 The free lunch is over，指明多核和并行程序才是提高程序性能的唯一办法
- 多进程的缺点
  - 太笨重，不好管理
  - 太笨重，不好切换

多线程概念

- 一个程序可以包括多个子任务，可串/并行
- 每个子任务可以称为一个线程
- 如果一个子任务阻塞，程序可以将CPU调度另外一个子任务进行工作。这样CPU还是保留在本程序中，而不是被调度到别的程序(进程)去。这样，提高本程序所获得CPU时间和利用率。

多进程和多线程对比

- 多进程 vs 多线程
  - 线程共享数据
  - 线程通讯更高效
  - 线程更轻量级，更容易切换
  - 多个线程更容易管理

多进程和多线程示例

- 多进程执行：启动两个java.exe
- 多线程执行：只启动一个java.exe
- 示例

总结

- 总结
  - 了解多进程和多线程基础概念
  - 了解多进程和多线程的特点和适用场合

代码(1) ProcessDemo1.java


```
public class ProcessDemo1 {

    public static void main(String[] args) {
        while(true)
        {
            int a = (int) (Math.random() * 100);
            System.out.println(" main thread is running " + a);
            try {
                Thread.sleep(5000); //5000毫秒
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

代码(2) ThreadDemo1.java


```
public class ThreadDemo1
{
    public static void main(String args[]) throws Exception
    {
        new TestThread1().start();
        while(true)
        {
            System.out.println("main thread is running");
            Thread.sleep(1000);
        }
    }
}
```

代码(3) TestThread1.java


```
class TestThread1 extends Thread
{
    public void run()
    {
        while(true)
        {
            System.out.println(" TestThread1 is running");
            try {
                Thread.sleep(1000); //1000毫秒
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
}
```

谢 谢！