守护线程和非守护线程

Java中的线程分为两种线程：用户线程（非守护线程）和守护线程。

守护线程是指在虚拟机运行的过程中为其他用户线程提供服务的一种线程。从这句话中我们可以清楚的明白看到，守护线程的存在是为其他用户线程提供服务的，因此，如果虚拟机中只有守护线程存在的时候，虚拟机就会杀死守护线程并退出，因为此时守护线程没有了服务的对象；反之，如果虚拟机运行当中还存在着非守护线程，那么守护线程就不会停止，并将一直运行下去，直到无非守护线程。

很好的一个例子就是java虚拟机的垃圾收集机制器，其实就是一个守护线程。而且要注意一定，在Daemon线程中创建的线程也是守护线程。

在线程start之前使用线程的setDaemon(true)方法，用户可以自定义守护线程。

|  |
| --- |
| public static void main(String[] args)  {  Thread thread1 = new Thread(new Runnable()  {  @Override  public void run()  {  for (int i = 0; i < 200; i++)  {  System.out.println("子线程"+i+"运行中");  }  }  });    Thread thread2 = new Thread(new Runnable()  {  @Override  public void run()  {  for (int i = 0; i < 2000; i++)  {  System.out.println("守护线程"+i+"运行中");  }  }  });    thread1.start();  thread2.setDaemon(true);  thread2.start();  System.out.println("主线程退出");  } |

多次运行会发现规律，守护线程可能并没有运行到2000次就结束了，但是子线程一定会运行200次，并且在主线程退出的时候，子线程和守护线程并没有结束；由此可以确定，守护线程会等待所有非守护线程结束后退出，也就是虚拟机中只剩下守护线程的时候随虚拟机退出，随时可能中断。

还有一点需要注意，守护线程并不是在所有非守护线程都结束时立马退出，而是有一个时间间隔，但是这个时间间隔很短，这是由线程在cpu上的运行特性决定的。