

# 多线程在网络程序设计中的应用<sup>\*</sup>

林 晓

( 华侨大学计算机科学系, 泉州 362011)

**摘要** 讨论多线程的思想在网络程序设计中的优越性, 同时论及多线程的基本原理、多线程与传统的多进程的区别和 Java 语言对多线程支持. 在网络中使用 Java 语言为开发工具, 利用多线程的思想进行网络通讯程序的设计, 以实现接收数据和发送数据的并行性. 所设计的程序简洁明了, 且易于实现. 以一个简单的网络通讯程序为例, 详细描述如何利用多线程的思想来设计网络通讯程序. 多线程在软件工程应用中具有非常积极的意义, 不仅可以用于网络程序设计, 也同样适用于其它方面的软件设计, 具有广阔的应用前景.

**关键词** 多线程, 并行执行, Java, 数据传输

**分类号** TP 393: TP 311.5

多线程编程是近几年提出的一种新的程序设计概念. 通过多线程来实现程序的并行执行, 既提高程序的性能, 又能增强程序的功能. 在传统的多用户、多任务操作系统, 如 UNIX, WINDOWS 95/NT 都可以同时执行多个程序称为多进程. 但是, 多进程也存在不少问题, 这主要表现在如以下几点. (1) 进程之间信息的交换非常复杂. (2) 进程与进程的切换涉及到多种资源, 管理开销大、耗时长. 每个进程在执行自己或其他进程的时候, 都要占用资源, 造成包括内存存在内的资源浪费. (3) 由于进程独享资源空间, 不利于数据与代码的共享, 使编程复杂化. 由于以上原因, 软件设计人员希望有一种新的程序设计方法, 用于解决进程编程所存在的问题. 在这种技术背景下, 线程就应运而生了.

## 1 线程的实现原理

多线程就是在程序中有多条执行路径, 这些不同的执行路径可以并行地工作. 线程的提出, 屏蔽了计算机并行执行的底层原理, 使程序设计人员从软件设计的角度来进行系统分析, 而不必去考虑其它的细节. 多线程编程相对于进程编程来说简单得多. 程序中线程的行为就像是函数, 可以通过全局变量交换信息, 也可以共享内在文件等资源, 与常规的编程方法类似. 在一个程序中, 可以利用线程来定义多个不同的行为, 让这些不同的行为能并行地工作. 其实, 在单个 CPU 的计算机中, 多个线程并不能真正地并行执行. 它是通过时间片机制, 或者当一个线程处于等待状态时, 自动切换到其他的线程, 可以实现多个线程的重叠执行. 在宏观上, 使用多个线程能并行执行, 仿佛是多 CPU 在同时工作.

## 2 多线程在网络程序设计中的应用

目前,一些较流行的开发语言,如 Visual C<sup>++</sup>, Borland C<sup>++</sup> 和 Java 等都支持线程编程。在网络日趋成为未来趋势的形势下,Java 语言的地位越来越突出,受到人们的重视。利用 Java 语言来编写网络应用程序,会使设计人员感到意想不到的简单与快捷。以下举一个类似于聊天的通讯程序,说明线程在网络程序设计中的优越性。

### 2.1 通讯程序的系统要求

在网上聊过天的人都知道,两个或多个人通过聊天程序进行聊天的时候,不可能等一个人说完了,另一个人才能说。也就是说,多个人同时通过聊天程序进行对话,各人的行为不会受到别人干扰,并且各人说的内容在屏幕上能实时地显示出来。所以,在进行系统设计时,必须通过多个线程来控制各人的发言与接收对方谈论的内容,使得各人所说的不受影响。考虑到程序的复杂性,现仅以两个人之间的通讯为例。假设,这两个人分别是 chatServer 和 chat-Client(两个独立的应用程序),分别简称 S, C。

对 S 来说,他必须能接收到 C 传过来的内容,即 C 通过键盘传过来的数据能在 S 的屏幕上实时的显示出来。而且, S 在看到 C 传过来的数据的同时,也可以输入自己所要讲的内容传给 C。同样, C 也要具备这两个要素。程序的原理框图如图 1 所示。从原理框图中可以看出, S 和 C

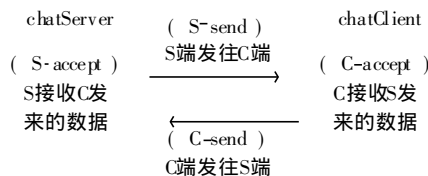


图1 通讯程序的原理框图

各需要两个线程。对 S 来说,他需要发送数据的线程(S-send),以及接收数据的线程(S-accept)。而对于 C 来说,他也需要发送数据的线程(C-send)及接收数据的线程(C-accept)。其实,线程(S-send)与线程(C-send)是执行同样的动作,而线程(S-accept)和线程(C-accept)也执行同样的动作。经过以上分析后,我们可以定义接收数据线程与发送数据线程。

### 2.2 通讯程序的线程代码

#### (1) 接收数据的线程为

```

class DispDataIn extends TextArea implements Runnable
{
    public DataInputStream in;
    Thread DspInThread;
    DisplayIn(DataInputStream in)
    {
        super( );
        setBackground( Color. gray);
        setEditable( false);
        this. in= in;
        DspInThread= new Thread( this);
        DspInThread. start();
    }
}
  
```

```

        public void run()
        {
            while(true)
            {
                try
                {
                    char ch = in.readChar();
                    appendText(String.valueOf(ch));
                }
                catch(Exception e)
                {
                    }
                }
            }
        }
    }

```

## (2) 发送数据的线程为

```

DispDataIn()
class DisplayDataOut extends TextArea implements
Runnable
{
    public DataOutputStream out;
    Thread DspOutThread;
    DisplayOut(DataOutputStream out)
    {
        super( );
        this.out = out;
        DspOutThread= new Thread(this);
        DspOutThread.start();
    }
    public void run()
    { }
    KeyDown()
    public boolean KeyDown(Event evt, int key)
    {
        try
        {
            out.writeInt(key);
        }
    }
}

```

```
        catch( Exceptione)
        {
        }
        return false;
    }
}
```

将以上两个线程放入 S 和 C 中, 使 S 和 C 都具有同时接收数据和发送数据的能力. 经网络连接后, S 和 C 就可以进行通讯. 由于本文主要描述线程在网络程序设计中的应用, 所以有关网络连接部分略去.

### 3 结束语

用线程来进行网络程序设计是一件自然、方便的事. 根据应用的要求, 让程序中的每个线程执行一定的动作, 将大大提高程序的性能. 除了在网络上的应用外, 多线程编程也可在多媒体、大型科学计算、通讯, 以及其它应用领域中发挥着积极的作用. 利用多线程来进行编程, 对于编写良好的应用程序是非常有意思的.

#### 参 考 文 献

- 1 姜 刚, 胡全星. Java 语言与程序设计. 北京: 人民邮电出版社, 1998. 219~249
- 2 刘甲耀, 严桂兰, 刘 波. Visual J++ 程序设计. 北京: 电子工业出版社, 1998. 171~173
- 3 Deitel H M, Deitel P J 编. Java 语言程序设计大全. 袁兆山等译. 北京: 机械工业出版社, 1997. 521~544
- 4 周 航, 蔡文胜, 李军北. Java 语言程序设计. 北京: 电子工业出版社, 1997. 124~150, 313~330
- 5 王克宏, 郁 欣, 姜 河等. Java 语言编程技术. 北京: 清华大学出版社, 1997. 63~74

## Application of Multi-Threading to Network Programming

Lin Xiao

(Dept. of Comput. Sci., Huaqiao Univ., 362011, Quanzhou)

**Abstract** A discussion is devoted to the advantages of multi-threading thinking in network programming; the fundamental principle of multi-threading; the difference between multi-threading and traditional multiple-process; and the support of Java language to multi-threading. The use of Java language as development tool and the application of multi-threading thinking to the programming of network communication will facilitate the realization of parallelism of accepting and transmitting of data and will lead to a concise and explicit program. A simple program of network communication is given as an example to describe how to use multi-threading thinking in designing program of network communication. The application of multi-threading to software engineering is of positive significance, it can be used in network programming and in software design of some other aspects, it has a vast range of prospects.

**Keywords** multithreading, parallel execution, Java, data transmission