

线程池在数字语音教学系统中的应用

Application of Thread Pool in Digital Speech Teaching System

丁亚军 万静华 (长沙 湖南师范大学计算机教学部 410081)

摘要:本文介绍了线程池的特点,数字语音教学系统的功能、特点及线程池在数字语音教学系统中的应用。

关键词:线程池 终端 网络 数字语音 SOCKET

1 引言

数字语音系统采用 C/S(客户端和服务端)模式,服务端即教师机或资源所在计算机,由计算机完成相关工作,要求能实现实时教学、同声传译、复听功能,支持视频、音频、文本资料点播等。客户端即学生端,采用语音终端。终端设备通过线程的方式与教师机相连。由于创建线程和释放线程是要消耗系统资源的,如果要完成一个工作要不停的创建和释放线程必然会造成很大的系统资源的浪费,学生端可能会经常出现上线、下线操作,势必浪费系统资源,所以采用线程池来实现学生终端设备与教师机的连接。

2 线程池简介

在面向对象的编程过程中,经常要创建和销毁对象,但其操作很费时,因为创建一个对象要获取内存资源或者其它系统资源,并且虚拟机还试图跟踪每一个对象,以便能够在对象销毁后进行回收。所以提高服务程序效率的一个手段就是尽可能减少创建和销毁对象的次数,尤其是一些很耗资源的对象创建和销毁。如何利用已有对象来服务,这是一个需要解决的关键问题,也就是一些“池化资源”技术产生的原因。

线程池的主要思想是先要创建一个容器来装这些已经创建的线程,在线程本次工作完成后,不释放线程,让线程等待。再有需要让线程去完成的工作,就把原来创建的线程取过来继续使用。这样节省了重复的创建释放线程的过程。但我们需要用一套机制来识别容器中的线程哪个是空闲的,哪个正在工作。我们所开发的学生终端设备只要与教师机连接成功后,所有曾登录的单播终端都存在于 UTP 中。登录后离线的终端不直接释放,便于重复利用。

3 线程池的实现

一个线程池至少应包含线程池管理器、工作线程、任务队列、任务接口等部分。其中线程池管理器(ThreadPool Manager)的作用是创建、销毁并管理线程池,将工作线程放入线程池中;工作线程是一个可以循环执行任务的线程,在没有任务时进行等待;任务队列的作用是提供一种缓冲机制,将没有处理的任务放在任务队列中;任务接口是每个任务必须实现的接口,主要用来规定任务的入口、任务执行完后的结束工作、任务的执行状态等,工作线程通过该接口调度任务的执行。

4 数字语音系统功能结构

4.1 数字语音系统功能划分

数字语音系统功能划分为两层:应用层(Application Layer - AL)和终端管理层(Terminal Management Layer - TML),AL 层结构如图 1 所示。

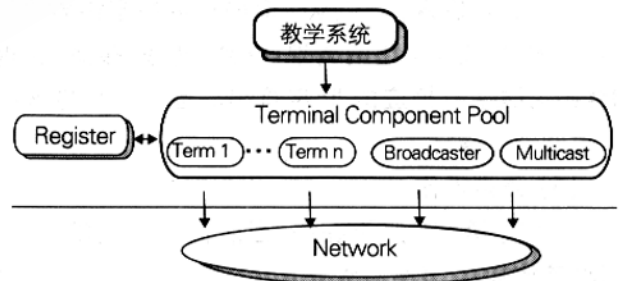


图 1 数字语音系统 AL 层功能结构

下面主要讨论终端管理层。TML 为直接与终端机沟通的底层服务。其原理是在 TML 中的终端组件(Terminal Component - TC),通过构筑于 UDP 协议之

上的通讯协议与终端机交互,对终端机的操作尽可能由 TC 完成。

4.2 终端管理层 (TML) 功能划分

4.2.1 终端管理

包括终端池维护、终端登录/注销、终端在线检测、终端检索、终端操作、终端事件监测等。

(1) 终端池维护。对单播终端,需要从 TCP 中申请终端时,首先根据申请者 IP 地址从 UTP 中检索,若未检索到终端且 TCP 中终端数量未超过 TCP 上限,则创建一个新的单播终端,初始化后向申请者回复响应并启动此终端;若获得的单播终端此时未非活动的,则向申请者回复响应并启动此终端;若获得的单播终端此时是活动的,则只向申请者回复响应(可考虑向申请者发送当前状态的信息 - 此情形适用于终端机断线后在超时前又再次登录)。响应中包括终端的各个端口等参数。终端被创建或重复使用时,将被关联一个关键字,便于后续检索。对单播终端,关键字为终端机的 IP 地址构成的无符号长整型,对多播终端,关键字为绑定的 socket 端口。

(2) 终端登录/注销。基于终端池维护以及终端事件监测。有终端请求登录时,从 TCP 中申请一个终端。如申请成功,则初始化终端。如初始化成功,则将其放置于 OTP 中,否则放置于 FTP 中。多播终端不存在登录/注销问题,一般由 AL 直接向 TML 申请。

(3) 终端在线检测。基于终端池维护以及终端事件监测。登录后被检测到离线的终端不直接释放,而回收 FTP 中,便于重复利用。多播终端不存在登录/注销问题,一般由 AL 直接通知 TML 回收。

(4) 终端检索。提供从 TCP 中检索指定终端的服务。检索时,应指定终端类型(在线、离线、多播)并提供终端关键字或相应的 IP 字符串。检索到相应终端时,返回一个指针。

(5) 终端操作。

① 状态查询:是否在线;当前状态(忙、闲、等待数据、等待响应、发生错误等)。

② 状态操作:复位、初始化、锁定、解锁、强制离线等。

③ 界面操作:由终端提供一个设备上上下文(Device Context - DC),AL 或 TML 在此 DC 中绘制界面,然后提交即可。

④ 流数据操作:数据同步传输、暂停、中止、快进、

快退、单遍或循环播放等。

(6) 终端事件监测。申请登录、在线状态、键盘事件、错误、流数据缓冲池状态等。

4.2.2 终端功能

根据终端的功能要求,其结构如图 2 和图 3。

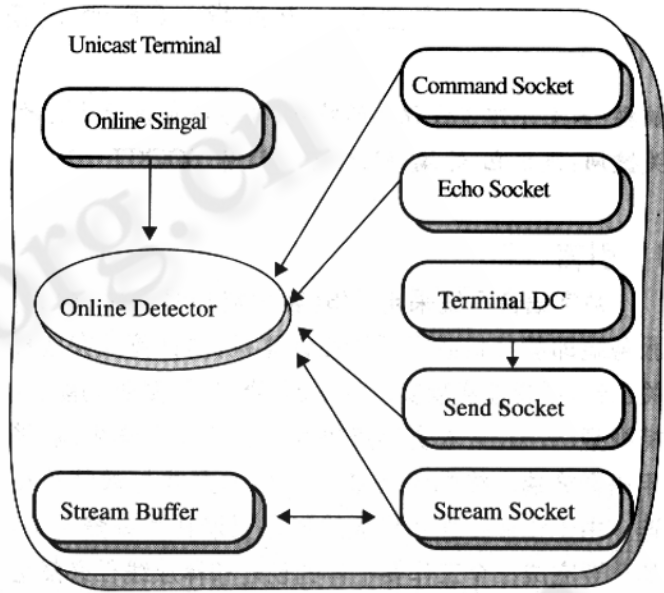


图 2 终端单播功能结构图

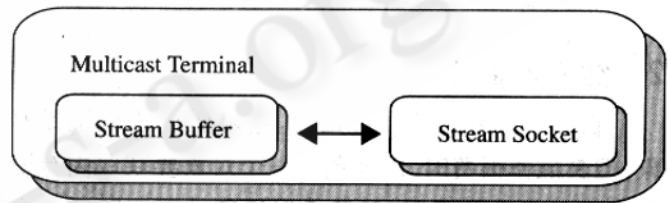


图 3 终端多播功能结构图

5 结束语

本文简要的介绍线程池技术及其应用。可以看出线程池技术对于服务器程序的性能改善是显著的。线程池技术在服务器领域有着广泛的应用前景。

参考文献

- 1 王华等, Visual C++ 6.0 编程实例与技巧[M], 机械工业出版社,1999,1。
- 2 Anthony Jones, Jim Ohlund. Windows 网络编程技术[M], 机械工业出版社,2000,3。