

多任务处理环境下的定时器应用算法

卢启衡^{1,3} , 冯晓红²

(1. 西南交通大学电气工程学院, 成都 610031; 2. 西南交通大学软件学院, 成都 610031; 3. 成都铁路运输学校, 成都 610081)

摘要: 本文通过对定时器原理的分析, 结合队列数据结构和网络协议中“时间戳”的概念, 提出了一种新的、较完善的多任务处理环境下的定时器应用算法, 并且给出了主要程序的流程框图。该算法具有较高的效率和可靠性, 其时间复杂度及空间复杂度均为 $O(n)$ 。

关键词: 多任务处理; 定时器; 算法

1 问题的提出

在面向对象的程序设计过程中, 经常会涉及到对定时器进行编程。在单定时任务处理时, 算法比较简单, 但对于多定时任务或复杂的嵌套定时任务的处理则要麻烦得多。目前的做法通常有两种: 使用一个定时器, 对不同的任务设置不同的标志, 在定时中断时, 根据不同的标志, 采取相应的操作, 但这种方法对嵌套定时任务无能为力; 使用多个定时器, 对不同的任务使用不同的定时器, 可该方法也无法解决任务数量或种类不确定的情况。而且不管是哪种方法, 都会在标志或定时器的管理上存在问题, 给程序的编制增加相当的难度。所以, 找到一种多任务处理环境下的通用定时器应用算法, 对于程序的简洁和可靠性具有重要的意义。

2 定时器原理

在面向对象程序设计中, 定时器是专门用于计时的一种控件。在设置了相应的时间间隔后, 开始倒计时, 当所设定的时间间隔一旦到达, 系统便会自动触发定时器事件来完成相应的操作, 同时, 系统自动恢复所设置的时间间隔, 开始下一轮的倒计时, 直至定时器的使能属性设置为假或时间间隔被修改为 0 时为止。

3 算法的描述

(1) 算法说明

由上述的定时器原理, 我们可以知道, 单定时任

务的处理是比较简单的, 但对于多定时任务或嵌套的定时任务, 处理起来就会复杂很多。实际上, 在多任务环境下, 定时器事件触发时, 如何区分不同的任务便成为解决这个算法的核心问题。为此, 笔者想到使用计算机网络协议中的经常用到的“时间戳”概念来解决该问题。

该算法的设计分为两个阶段: 算法准备阶段和算法运行阶段。在算法准备阶段 (应用系统初始化阶段), 首先设置一个整型全局变量 (初值可设为数据类型所允许的最大值) 来描述系统统一时间标志; 然后为定时器设置一个固定的时间间隔 (通常设置为 1 秒), 并启动定时器。在算法运行阶段 (应用系统正常运行后), 涉及到添加定时器任务的处理和定时器事件的处理。在添加定时器任务时, 将整型全局变量的当前值减去该任务所需的时间间隔作为其“时间戳” (到达时间); 而每次当定时器所设定的时间间隔到达, 触发定时器事件时, 先将前面所述的整型全局变量减 1, 再检查当前所有定时器任务的“时间戳”, 若与某个任务的“时间戳”相同, 则执行该任务。另外, 为了配合此算法, 还应设置两个全局数组来分别存储所有的定时器任务的“时间戳”和“任务描述”。

(2) 算法流程图

变量说明

`gviGeneral`: 整型全局变量, 初值可设为数据类型允许的最大值, 用于描述系统统一时间标志。在固定

的时间间隔到达并触发定时器事件时,其值递减 1。

`gviTask[MaxTaskNum]`: 整型全局数组, 存储各个定时器任务的“时间戳”的队列。

`gvsTask[MaxTaskNum]`: 字符串全局数组, 存储各个定时器任务简要描述信息的队列。

`viInterval`: 整型局部变量, 存储各个定时器任务所需的时间间隔。

定时器任务添加流程图(如图 1)

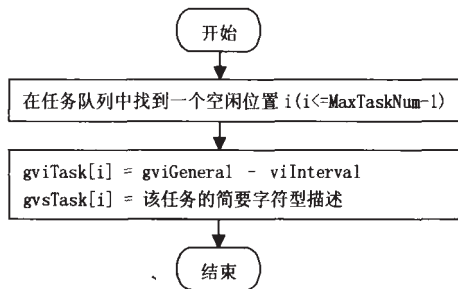


图 1

定时器事件处理流程图(如图 2)

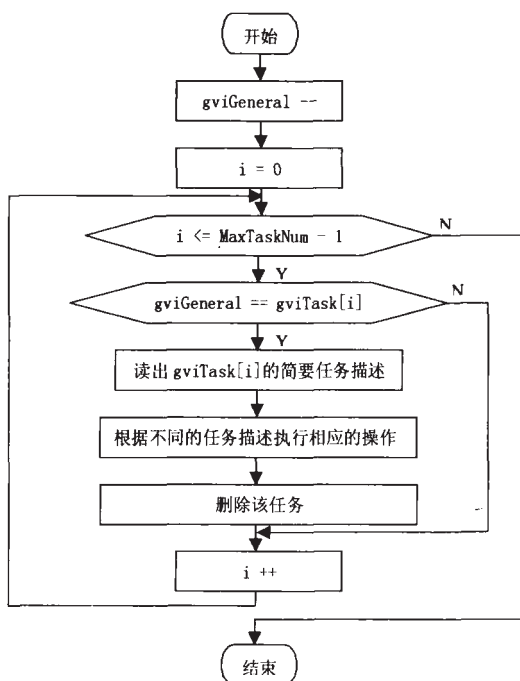


图 2

在任务队列中查找空闲位置及任务删除程序的结构相对比较简单, 鉴于篇幅所限, 本文不再给出具

体流程图, 读者可自行补充。另外, 需要注意的是, 在设置时间间隔时, 应将扫描一遍所有任务队列所需的时间控制在所设定的固定时间间隔范围之内, 否则, 可能会造成任务丢失的情况产生。

4 算法分析

该算法用到了队列数据结构, 而从整个算法的结构上分析, 其空间复杂度和时间复杂度均为 $O(n)$ 。这里的 n 就是最大任务数 `MaxTaskNum`, 所以, 该算法的效率是比较高的。

结 语

本文通过对定时器编程原理的分析, 结合数据结构中的队列和网络协议中的“时间戳”的概念, 提出了一种新的定时器编程算法。该算法比较完善地解决了多任务处理环境下的定时器编程问题, 并且实现了较高的效率。另外, 为了使整个算法简化, 在每一次插入定时器任务时, 并没有对当前所有任务按“时间戳”进行排队, 而是将新任务直接插入到一个空闲的位置; 同时, 在处理任务时, 也没有严格按照队列所要求的“先入先出”的原则处理, 而是进行全程扫描, 处理所有满足到达时间的任务。这样做虽然增加了队列扫描时间, 但节省了排队时间, 并且在多个任务具有同一“时间戳”时, 能够避免任务丢失的情况产生。相信随着面向对象编程技术的不断发展, 该算法会得到越来越多的应用。

参考文献

- [1](美) Andrew S. Tanenbaum. Computer Networks (Third Edition). 熊桂喜, 王小虎(译). 北京: 清华大学出版社, 2003
- [2] 谭浩强. C 程序设计. 北京: 清华大学出版社, 1999
- [3] 苟帅, 张俊平. C++Builder 5.0 程序员指南. 北京: 希望电子出版社, 2000
- [4] 杜秋华, 康慧芳. 可视化编程应用 - Visual Basic 6.0. 北京: 人民邮电出版社, 2004
- [5] 严蔚敏. 数据结构. 北京: 清华大学出版社, 2004
- [6] 王晓东. 算法设计与分析. 北京: 清华大学出版社, 2004

(收稿日期: 2006- 11- 20)

(英文摘要见下页)

(接上页)

Timer Application Algorithm under Multitasking

LU Qi-heng^{1,3} , FENG Xiao-hong²

- (1. Electric Engineering College in Southwest Jiaotong University, Chengdu 610031 China;
2. Software Engineering College in Southwest Jiaotong University, Chengdu 610031 China;
3. Chengdu Railway Transportation School, Chengdu 610081 China)

Abstract: After analyzing the working principle of timer and with the help of the concepts of queue (Data Structure) and time stamp (Computer Network), the article gives a new algorithm to solve problems in the applications of timer under multitasking. The flow charts of some main programs are also provided in the article. The algorithm has a efficient space and time complexity, which are both at $O(n)$.

Key words: Algorithm; Multitasking; Timer



(上接第 52 页)

A Study on the Development of Enterprise Application Integration based on Common Information Model

DU Li-feng , LI Bing , LIANG Guo-jian , LI Jia-ming

(Zhongshan Power Supply Bureau, Zhongshan 528400 China)

Abstract: To eliminate information islands of existing application systems, Zhongshan Power Supply Bureau develops a common information model on enterprise level, keeping to IEC61970 and IEC61968 international standards. The Bureau designs an application integration scheme to meet the actual requirement in data exchange between different application systems. The application integration scheme based on common information model endows enterprises with sustainable development ability and gives the universal reference value for different industries.

Key words: Enterprise Application Integration; Common Information Model; Application Integration Architecture