A Teaching System about Computer Applications Foundation Based on Scheduling Algorithm

HUANG Man-ying
Guangxi Economic Management Cadre College, nanning,China
e-mail：9365282@qq.com

Abstract: Scheduling algorithm-based teaching system about computer applications foundation is a soft system supporting computer applications foundation teaching. Students can learn knowledge about computer applications foundation by using the soft under the guidance from the teacher. In a real-time system, an event must be answered within a limited time, so the first step is to realize real-time scheduling. When n users log in the system and send out requests, the system must can allocate time to every user to use the processor effectively. To be fair, a scheduling algorithm based on time slice circular is used. The scheduling program allocate a time slice to evry task before beginning scheduling. Tasks are sorted by First In First Out, every task use the processor by turns and the time duration is the length of the time slice, the task is moved to the end of the line when the running of the time slice is complete, and so on, till all tasks are completed.

Keywords: scheduling algorithm; computer applications foundation; teaching system

基于调度算法的计算机应用基础教学系统研究

黄曼莹
广西经济管理干部学院，广西南宁，中国，530007
E-mail：9365282@qq.com

【摘要】基于调度算法的计算机应用基础教学系统是支持计算机应用基础教学的软件系统。它是基于计算机应用基础的教学内容，由教师作为导向，面向学生的实时软件系统。实时系统要对现实的事件在一个限定的时间内响应，基础问题是解决实时调度问题。当有 n 个用户登录系统后，发出请求任务，需要合理有效分配用户使用处理器的时间。为了体现公平性，使用时间片轮转调度算法进行用户请求任务调度。基于时间片轮转的调度算法，调度程序在调度开始前分配给每项任务一个时间片，任务按照先进先出的队列进行排序，从第一项任务开始，轮流使用处理器，每一项任务使用处理器的时间是时间片的长度，时间片运行结束，就把任务移到队列末，直到所有任务执行结束。

【关键词】调度算法；计算机应用基础；教学系统；研究

1 引言

计算机应用基础是一门实践性很强的课程，教师在教学中，普遍使用的教学方法有“案例教学法”、“任务驱动法”等。这两种方法被认为是缩短了教学与实践的距离的较好的教学方法。由于多媒体教室的局限性，导致这些教学方法不能施展身手。在教学过程中，教师处于“独唱”局面，学生学习处于被动局面。基于调度算法的计算机应用基础教学系统，目的在于提高学生学习积极性，变被动为主动，增强学生学习效率，以满足各行各业对所需人才的计算机应用基础知识掌握的高要求。

基金项目：广西区教育厅科研项目（200911Lx543）
Project sources：scientific research project of the Education Department of Guangxi（200911Lx543）

2 平台构建

基于调度算法的计算机应用基础教学系统是支持计算机应用基础教学的软件系统，它是基于计算机应用基础的教学内容，由教师作为导向，面向学生的软件系统。计算机应用基础教学系统是一个提供教师进行教学，学生进行学习，实现师生互动的平台。

学生用户登录系统，系统进行身份认证，认证通过

后，学生可以进行学习内容选择（按照模块选择或者是搜索查询选择）。学习内容的更新，需要在后台进行数据维护。

该教学系统实现学生的分层次学习。把学习内容分成三个模块，基础模块、拓展模块和高级模块，如“Figure 1”（图 1）所示。

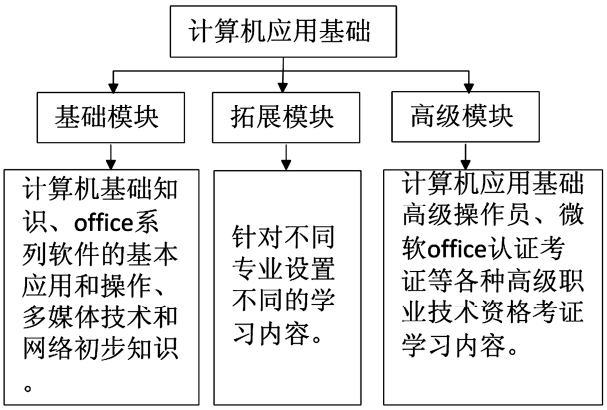

Figure1. System functional framework
图 1.系统功能框架

基础模块为必须掌握的基本知识，拓展模块按照不同专业设置不同的拓展任务，以会计电算化等几个专业的 Excel 学习内容为例，如 Table 1（表 1）所示，由于学生还须面临每年全国高校的联考，基础模块与拓展模块内容包含所有考试大纲内容，保证考试通过率。对于学有余力的学生，还可以跨专业进行学习，这和我们国家推行的“通才”教育相一致，学生知识面也会得到扩展。高级模块则是专门针对各种高级职业技术资格考证（计算机应用基础高级操作员、微软 office 认证考证等）所设置的内容。

Table 1. Extended contents
表 1.拓展内容

专业	拓展内容
会计电算化	“员工工资表”、“固定资产折旧表”等
物流管理	“物料计划表”、“仓储出货报表”等
人力资源	“员工任职能力匹配分析表”、“业务人员绩效跟踪评估表”等

2.1 系统模型构建

2.2.1 构建思路

基于调度算法的计算机应用基础教学系统的理论基础为调度算法。调度算法用于教学系统教学资源调度、教学模式调度等。本教学系统基于学生登录系统的进程来研究调度算法在教学系统中应用。

2.1.2 系统模型

当调度实时任务时，要考虑计算任务的执行成本。任务的执行成本为处理任务所需要的时间。

定义 1：当任务执行时间小于时间片 P_BaseTime，则任务执行成本=任务执行的时间。

定义 2：任务执行时间大于时间片 P_BaseTime，任务的执行成本=前一任务的运行时间结果 P_EndTime +上一次调度后剩余的时间 P_RemainTime[i]。

定义 3：当任务的剩余 P_RemainTime[i]≠0 时，下一周期继续进行调度，调度次数 P_Count[i]累计加 1。

定义 4：N 项任务的平均响应时间为所有任务的执行时间的平均值 avg。

2.2 基于调度算法的实时系统模式

2.2.1 调度算法

实时系统是指在某一限定的时间内，对外部现实事件的响应。在实时系统中，合理安排各个任务的执行过程，称之为任务调度。每一项任务在处理器上开始执行的时间与执行时间的长短是不同的，不论何时发生，不论执行时间的长短，所有的调度工作都是按照原本设置好的算法来进行。

(1) 调度算法分为抢先调度和非抢先调度

抢先调度，首先给任务分配一个优先级别，进行调度的时候首先执行优先级别高的任务，以此类推，直至任务执行完毕。若在执行任务过程中，又出现优先级别高于正在执行的任务时，则终止当前任务，重新执行调度，跳到优先级别高的任务继续执行，这样的调度算法，定义为抢先调度；非抢先调度，在执行任务过程中，不会终止当前任务，而是一直执行直到完成任务。当一个紧急任务到达时，不能有效进行调度，导致系统性能恶化。

（2）动态调度和静态调度

动态调度，指任务在执行过程中，被调度的时间是不确定的，是实时的，这样的调度称为动态调度。使用动态调度，处理器保持对任务的跟踪，当出现优先级别更高的任务，则可以实现前线调度；静态调动，按照任务优先级别执行，一旦开始执行某个任务，其他任务执行的时间都按照原来设置的顺序执行。如果在调度系统

978-1-935068-31-0 © 2010 SciRes.

中增加或者删除处理器，都必须停止当前任务，重新进行调度，因此，静态调度在适应性上灵活性不高。

2.2.2 调度算法的系统模式

应用于软件系统的实时调度算法有三种类型，分别是基于优先级别的调度算法、基于比例共享的调度算法和基于时间进程的调度算法。

当多个用户登录系统，为了保证每个登录系统的用户分配到的访问系统的时间相同，在计算机应用基础教学系统中选择基于比例共享的调度算法来进行任务调度。比例共享调度算法有时间片轮转法、公平共享等几种类别。

假设在任务进行调度的过程中，每个登录系统的用户分配的处理器的时间相同，则认为调度实现了公平调度。时间片轮转法是简单又公平的进程调度算法。

时间片指分配给每一任务在处理器上运行的时间，一般设置为 100ms。它的工作原理是：每一任务以自己分配到的时间片来使用处理器。如果时间片到期时，即使任务还没完成，也必须停止执行，系统将会重新进行调度，另一任务继续使用处理器，以此类推，直到任务完成。调度程序在调度开始前先先把一组任务进行队列排序，第一项任务的时间片运行结束之后，就把该任务移到队列末尾，处理器继续把时间让给新的第一项任务，每一项任务都轮流使用处理器的时间，且由于时间片的限制，使用 cpu 的时间都相同，如“Figure2”（图 2）所示。

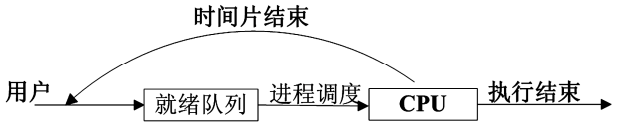

Figure 2. Flow Process of Scheduling Algorithm
图 2.调度算法流程

用 c#语言来模拟实现对 n 个任务采用时间片轮转调度算法的调度。

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace 时间片轮转模拟进程调度算法
{    class Program
    {    static int P_EndTime = 0;//存放进程本次结束时的时间
     static void Main(string[] args)
{    int PStates = 1; int i = 0;//计数器
     int avg = 0;
     Console.Write("请输入一个 1 到 100 之间的整形进程数 P_Number:");
     int P_Number = Convert.ToInt32 (Console.ReadLine());
     int[] P_RemainTime = new int[100];//存放进程的剩余时间
     int[] P_Count = new int[100];//存放进程调度次数
     Console.Write("\n 请输入标准时间片 P_BaseTime 大小:");
     Int P_BaseTime=convert.ToInt32(Console.ReadLine());
    Console.WriteLine("\n 请依次输入各个进程的服务时间");
    for ( i = 0; i < P_Number;i++)
   {   P_RemainTime[i] = Convert.ToInt32(Console.ReadLine());
       avg = avg + P_RemainTime[i];P_Count[i] = 0;
    }
      avg=avg/P_Number;
   Console.WriteLine("被调度进程\t 进程调度次数 \t 本次运行时间结果\t 剩余时间");
    while (PStates == 1)
{
    for ( i = 0; i < P_Number; i++)
   {
      if (P_RemainTime[i] != 0)
     {
       if (P_RemainTime[i] >= P_BaseTime)
      {    P_RemainTime[i] -= P_BaseTime;
           P_EndTime+= P_BaseTime;
           P_Count[i] = P_Count[i] + 1; Console.WriteLine
           ("\t{0}\t\t{1}\t\t{2}\t\t{3}", i + 1,
           P_Count[i], P_EndTime,
```

```
                P_RemainTime[i]); }
            else
            {    P_EndTime = P_EndTime +
                 P_RemainTime[i];
                 P_Count[i] = P_Count[i] + 1;
                 P_RemainTime [i] = 0;
                 Con-
                 sole.WriteLine("\t{0}\t\t{1}\t\t{2}\t\t{
                 3}", i + 1, P_Count[i], P_EndTime,
                 P_RemainTime[i]
                 );
            }
          }
          else
         { continue;
          }
      }
      for ( i = 0; i < P_Number; i++)
     {if (P_RemainTime[i] != 0)
       {PStates = 1; break;}
        else
       {continue; }
     }
    if (i >= P_Number)
   {PStates = 0;}
}
          Console.WriteLine("\n 所有进程执行时间的
              平均值:"+avg.ToString());
          Console.WriteLine("\n 进程全部运行完成!");
          Console.ReadLine();
              }
      }
 }
```

3 模拟测试结果输出

实验环境：实验操作系统 windows server2003、SQL 2005，测试环境采用 C#2.0 构建。实验条件设计：实验输入进程数为 5 个为一组，设定的时间片大小为 100ms，5 个进程的服务时间分别为：55ms，68ms，123ms，209ms，98ms。经过对 1000 次不同数据测试结果分析，执行时间<100ms 的进程先来先完成，在其他执行进程时间不同时，执行成本相同，执行时间>100ms 的进程在其他执行进程时间不同时，执行成本均不同。对实验组的 1000 次测试结果的输出平均值如 Table 2（表 2）所示：

Table 2. Task execution cost
表 2.任务执行成本

任务	执行时间（ms）	执行成本（ms）
1	55	55
2	68	68
3	123	321
4	209	330
5	98	98

运行结果为:进程 1 调度 1 次,运行时间结果为 55ms,剩余时间为 0；进程 2 调度 1 次,运行时间结果为 123 ms,剩余时间为 0；进程 3 调度 2 次,第一次运行时间结果为 223ms,剩余时间为 23，第二次运行时间结果为 444ms,剩余时间为 0；进程 4 调度 3 次,第一次运行时间结果为 323ms,剩余时间为 109，第二次运行时间结果为 544ms,剩余时间为 9，第三次运行时间结果为 553ms,剩余时间为 0；进程 5 调度 1 次,运行时间结果为 421ms,剩余时间为 0。所有进程执行时间的平均值为 110ms.模拟实验结果如“Figure 3”（图 3）所示。

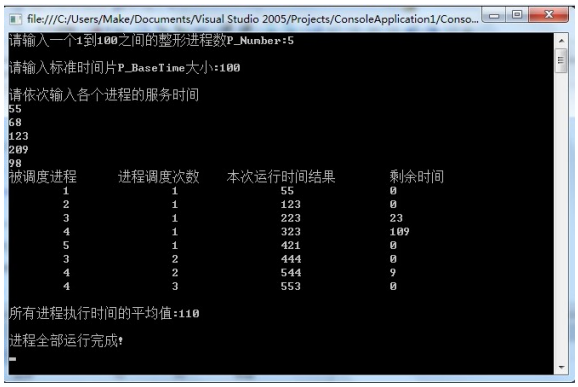

Figure 3. Result of simulation trial
图 3.模拟实验结果

4 结论

实时系统要对现实的事件在一个限定的时间内响应，基础问题是解决实时调度问题。某一任务可以使用处理器的开始时间和执行时间，用不同的调度算法就得到不同的结果。基于时间片轮转的调度算法是即公平又简单的任务调度算法。并且该算法可以灵活添

978-1-935068-31-0 © 2010 SciRes.

加任务，进行硬件扩充，系统的适应性较好。

References (参考文献)

[1] Stephanie L. Brooke, Using the Case Method to Teach Online Classes: Promoting Socratic Dialogue and Critical Thinking Skills[J]. International Journal of Teaching and Learning in Higher Education. 2006,(2) : 142-149.

[2] H.-G.Yook, Myung-Soon Park, Scheduling GEN_BLOCK Array Redistribution, The Journal of Supercomputing. 2002,3(22) : 251-267

[3] Chiussi,F.M,Francini,A,Khotimsky,D.A,Krishnan,S.Feedback control in a distributed scheduling architecture. Global Telecommunications Conference. 2000. GLOBECOM'00. IEEE. 2000(1) : 525–531.

[4] L. Tancevski, A. Ge, G. Castanon, L. Tamil, A New Scheduling Algorithm for Asynchronous, Variable Len- gth IP Traffic Incorporating Void Filling, Proceedings, Optical Fiber Communication Conference (OFC). 1999,2 (3) : 180-182.

[5] Y. Xiong, M. Vanderhoute, H.C. Cankaya, “Control Architecture in Optical Burst-Switched WDM Networks,”IEEE Journal on Selected Areas in Communications. 2000,(18):1838-1854.

[6] A. Neukermans, R. Ramaswami, “MEMS Technology for Optical Networking Applications,” IEEE Communica-tions Magazine. 2001,(39) : 62-69.