**操作系统课程设计**

**读书工程报告**

**学期 2014-2015学年第一学期**

**学院 计算机科学技术学院**

**学号 20123422**

**姓名 邸富赓**

**2014年 12月 25日**

一 基本理论阐述

1. 进程

并发程序和顺序程序有本质上的差别，为了能更好地描述程序的并发执行，实现操作系统的并发性和共享性，引入“进程”的概念。进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动，是系统进行资源分配和调度的一个独立单位。处理器是计算机系统中最重要的资源。在现代计算机系统中为了提高系统的资源利用率CPU将为某一程序独占。通常采用多道程序设计技术即允许多个程序同时进入计算机系统的内存并运行，我们今天使用的操作系统主要是分时系统，由调度程序调入的多个作业共享 CPU 资源，其中每个作业只执行极短的一段时间，比如 0.1s，我们称为一个时间片极短的时间过后暂停执行，调入下一个程序。这样在不长的一段时间内比如 5s内有限的进程少于50个都可以得到至少一次的执行用户请求可以得到及时的响应。这种作业调度的方式我们称为时间片轮转法。这种执行的方式叫并发执行并发性也是分时系统的基本特性之一。 在分时系统中为了执行一项作业就需要把要执行的作业程序载入内存中作为程序段为作业分配相应的数据空间作为数据段，并加入一个控制块PCB用来保存当前作业执行所必须的一些信息使之能够并发执行。内存中的程序段、数据段和 PCB 我们称为一个进程实体，而一个进程实体的执行过程我们称为进程。

2.进程的三种基本状态

1.进程的三种基本状态

进程在运行中不断地改变其运行状态。通常一个运行进程必须具有以下三种基本状态。就绪(Ready)状态，当进程已分配到除CPU以外的所有必要的资源，只要获得处理机便可立即执行，这时的进程状态称为就绪状态。执行（Running）状态当进程已获得处理机，其程序正在处理机上执行此时的进程状态称为执行状态。阻塞(Blocked)状态正在执行的进程，由于等待某个事件发生而无法执行时，便放弃处理机而处于阻塞状态。引起进程阻塞的事件可有多种。例如等待I/O完成、申请缓冲区不能满足、等待信件(信号)等。

2.进程三种状态间的转换一个进程在运行期间不断地从一种状态转换到另一种状态，它可以多次处于就绪状态和执行状态也可以多次处于阻塞状态。图3\_4描述了进程的三种基本状态及其转换。(1) 就绪→执行处于就绪状态的进程当进程调度程序为之分配了处理机后该进程便由就绪状态转变成执行状态。(2)执行→就绪 处于执行状态的进程在其执行过程中，因分配给它的一个时间片已用完而不得不让出处理机于是进程从执行状态转变成就绪状态。(3)执行→阻塞正在执行的进程因等待某种事件发生而无法继续执行时便从执行状态变成阻塞状态。(4)阻塞→就绪 处于阻塞状态的进程若其等待的事件已经发生于是进程由阻塞状态转变为就绪状态。

3、程序和进程程序是存储在外存储器、静态的为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。进程是程序运行时调入内存运行的形态是一个正在执行的程序。进程是动态的运行在内存中。程序是指一套软件它由多个文件相互配合来运行给你完成工作。比如你打开网站运行的就是IE浏览器程序进程是相对系统而言现从DOS系统以后windows系统都是多任务系统，也就是它可以同时处理多样事务也就是可以同时运行多个程序一个程序也可以运行多次每个程序运行一次都必须占用一个以上进程比如你打开一个网页窗口该程序就占了一个进程如果你打开多个网页窗口IE就会启运多个进程。这些都可以打开“任务管理器”“进程”中查看的到。你结束了该进程相应的软件也就会关闭。

4、进程分类进程一般分为交互进程、批处理进程和守护进程三类。值得一提的是守护进程总是活跃的一般是后台运行守护进程一般是由系统在开机时通过脚本自动激活启动或超级管理用户root来启动。比如在Fedora或Redhat中我们可以定义httpd 服务器的启动脚本的运行级别此文件位于/etc/init.d目录下文件名是httpd/etc/init.d/httpd 就是httpd服务器的守护程序当把它的运行级别设置为3和5时当系统启动时它会跟着启动。[root@localhost ~]# chkconfig --level 35 httpd on 由于守护进程是一直运行着的所以它所处的状态是等待请求处理任务。比如我们是不是访问 LinuxSir.Org LinuxSir.Org 的httpd服务器都在运行等待着用户来访问也就是等待着任务处理。

5.进程的属性

进程具有如下的属性. (1)用户所有 进程为用户所有,其可为进程提供安全措施. (2)最少有一个线程 每个进程至少有一个线程,进程创建的同时会创建一个线程 (3)拥有句柄 拥有的对象都有其句柄. (4)可继承句柄，有些共有的属性可被继承,利用Dupulicate Handle 函数或在创建时使对象可被继承来改变继承标识.例如,管道是读和写句柄,在管道创建时,它就可以被指定成可继承的. 2.进程的资源 进程拥有一系列的系统资源,比文件,管道,通信端口以及在进程被执行时由操作系统分配的信号量.典型的程序资源包括打开的文件,信号量和动态分配的内存.这些资源的标志都保存在进程的对象句柄表中. 一个进程可访问所有系统资源,所以一个进程也可以利用操作系统提供的资源管理器.进程可以访问的系统资源,所以一个进程也可以利用操作系统提供的资源管理器.进程可访问的系统资源如下所述. (1)访问标志 一个访问标志被操作系统附加在一个进程中.一个进程可得到其标志信息或改变其某些属性.例如.该进程具有保密特权,该进程必须先对它标志. (2)虚拟地址空间 隔离地址空间按进程的需要被指定并被虚拟存储管理器(VMM)跟踪.如果没有有效的内存,指定内存请求就会失败.一上进程不能直接改变包含自己虚拟地址的记录.所有的更改都必须间接地经过VMM. (3)对象句柄 一个进程可以被各种对象和资源拥有,创建或继承句柄.这些句柄保存在进程所拥有的对象表中.对象管理器保存了一个对象表,该表有其进程所拥有的对象句柄清单.例如,线程,文件,同步对象和共享内存. (4)资源引用限制 每个进程对其线程可为对象打开句柄的内存数量都有限制.这个限制叫作资源引用限制,它确保一个进程不能独占甩有的资源. (一个进程利用对象管理器来改变或得到有关它的访问标志或对象表的信息;利用虚拟存储管理器(VMM)跟踪进程的虚拟地址.) 3.进程与线程的相关性 一个线程可以看作是操作系统分配给CPU时间(即时间片)的基本单位.一个线程可以执行应用程序(可看作进程实例)代码的任何部分,该部分可以是另一线程正在执行的代码. 操作系统创建了进程之后,同时会创建一个线程执行进程中的代码.通常把这个线程称为该进程的主线程,主线程运行时,也可以创建其他线程.一个进程的所有线程共享虚地址空间.全局变量以及该进程的操作系统资源. 一个进程提供了程序运行所必需的一切,单线程同样如此.该默认进程(主线程)执行程序代码.一个进程中的主线程从一个入口点开始(对windows程序而言是WinMain()函数),然后根据代码的循环,条件,函数调用等情况继续执行.当进程被终止时,主线程同时也被终止. ( 同一个程序,可由多个进程执行.而所有进程都必须至少有一个线程,线程本身基本上不拥有系统资源,它与同属一个进程的其它线程共享进程所拥有的全部资源).

6. 父进程和子进程

他们的关系是管理和被管理的关系，当父进程终止时子进程也随之而终止。但子进程终止父进程并不一定终止。比如httpd服务器运行时我们可以杀掉其子进程父进程并不会因为子进程的终止而终止。在进程管理中当我们发现占用资源过多或无法控制的进程时应该杀死它以保护系统的稳定安全运行。

二当前理论或实践应用现状

1.线程、SMP 和微内核在许多操作系统中传统的进程概念被分为两部分一部分负责管理资源所有权另一部分负责指令流的执行。一个单独的进程可包含多个线程。使用多线程的组织方法对程序的结构化和性能方面都有很大的帮助。SMP 是一个拥有多处理器的计算机系统其中的每一个处理器都可以执行所有应用程序和系统代码。SMP 的组织方法增强了系统的性能和可靠性。SMP通常和多线程机制一起使用即使没有多线程机制也能很大幅度的提高系统性能。微内核是操作系统为了减少运行在内核模式的代码量的一种设计方式并且分析了这种方法的优点。

2.并发互斥和同步相交进程之间的关系主要有两种同步与互斥。所谓互斥是指散步在不同进程之间的若干程序片断当某个进程运行其中一个程序片段时其它进程就不能运行它们之中的任一程序片段只能等到该进程运行完这个程序片段后才可以运行。所谓同步是指散步在不同进程之间的若干程序片断它们的运行必须严格按照规定的某种先后次序来运行这种先后次序依赖于要完成的特定的任务。显然同步是一种更为复杂的互斥而互斥是一种特殊的同步。也就是说互斥是两个线程之间不可以同时运行他们会相互排斥必须等待一个线程运行完毕另一个才能运行而同步也是不能同时运行但他是必须要安照某种次序来运行相应的线程也是一种互斥。互斥是指某一资源同时只允许一个访问者对其进行访问具有唯一性和排它性。但互斥无法限制访问者对资源的访问顺序，即访问是无序的。同步是指在互斥的基础上大多数情况通过其它机制实现访问者对资源的有序访问。在大多数情况下同步已经实现了互斥，特别是所有写入资源的情况必定是互斥的。少数情况是指可以允许多个访问者同时访问资源。

3. 并发死锁和饥饿，死锁是这样的一种情况，一组进程中的两个或多个进程要等待该组中的其他成员完成一个操作后才能继续运行但是没有成员可以继续。死锁是一种很难预测的现象并且没有比较容易的通用解决方案。处理死锁的三个主要手段是预防、避免和检测。饥饿时一个准备运行的进程由于其他进程的运行而一直不能访问处理器的情况。从打的方面来说饥饿是被当成调度问题来处理的。

4.进程调度，我们经常遇到两个或多个进程，例如，生产者和消费者，在逻辑上均可以运行的情况。当有多个进程就绪时操作系统必须决定先运行哪一个。操作系统中作出这种决定的部分称作调度程序(scheduler)它使用的算法称作调度算法(scheduling algorithm)。再回到早期以磁带上的卡片映像作为输入的批处理系统时代，那时的调度算法很简单依次运行磁带上的下一个作业。对于分时系统则调度算法要复杂一些因为经常有多个用户等待服务而且同时可能存在多个批处理流，例如，保险公司理赔。即使在个人电脑上也可能有若干用户启动的进程竞争CPU更不要说还有后台作业例如网络或收发电子邮件的精灵进程。在讨论具体的调度算法之前我们应该考虑一下调度程序要达到的目标。毕竟调度程序关注的是确定策略而并不提供机制。一个好的调度算法应当考虑很多方面其中可能有1 公平 - 确保每个进程获得合理的 CPU 份额。 2 有效 - 使 CPU 百分之百地忙碌。3 响应时间 - 使交互用户的响应时间尽可能短。4 周转时间 - 使批处理用户等待输出的时间尽可能短。5 吞吐量 - 使每小时处理的作业数最多。进程的创建——调用进程创建原语creat步骤1申请空白PCB、2分配资源、3初始化进程控制块 4插入就绪队列进程的终止Termination of Process 进程的阻塞与唤醒——调用进程阻塞原语block和唤醒原语wakeup进程的挂起与激活——调用进程挂起原语suspend和激活原语active进程同步——主要任务是对多个相关进程在执行次序上进行协调。 信号量机制1整型信号量 2记录型信号量3AND型信号量AND同步机制将进程在整个运行过程中需要的所有资源一次性的全部分配给进程待进程使用完后再一起释放。 4信号量集——AND信号量机制的扩充信号量的应用1实现进程互斥 2实现前趋关系管程机制 管程——代表共享资源的数据结构以及由对该共享数据结构实施操作的一组过程所组成的资源管理程序共同构成了一个操作系统的资源管理模块。管程被请求和释放资源的进程所调用。进程通信类型1共享存储器系统Shared-Memory System——包括基于共享数据结构和存储区的通信方式。2消息传递系统Message passing system 3管道通信——管道用于链接一个读进程和一个写进程以实现它们之间通信的一个共享文件pipe文件。消息传递通信的实现方法1直接2间接，通过私有或公有或共享信箱收发消息，进程同步方式1汇合rendezrous发送进程阻塞接收进程阻塞，主要应用于进程之间紧密同步2发送进程不阻塞，接收进程阻塞应用最广3发送进程和接受进程均不阻塞常见，本地进程之间的通信主要应用消息缓冲队列通信机制消息缓冲区数据结构，一般来说,我们把正在计算机中执行的程序叫做"进程"(Process) ,而不将其称为程序(Program)。所谓"线程"(Thread),是"进程"中某个单一顺序的控制流。新兴的操作系统,如Mac,WindowsNT,Windows 95等,大多采用多线程的概念,把线程视为基本执行单位。线程也是Java中的相当重要的组成部分之一。甚至最简单Applet也是由多个线程来完成的。在Java中,任何一个Applet的paint()和update()方法都是由Awt(Abstract Window Toolkit)绘图与事件处理线程调用的,而Applet 主要的里程碑方法——init(),start(),stop()和destory() ——是由执行该Applet的应用调用的。 单线程的概念没有什么新的地方,真正有趣的是在一个程序中同时使用多个线程来完成不同的任务。某些地方用轻量进程(Lightweight Process)来代替线程,线程与真正进程的相似性在于它们都是单一顺序控制流。然而线程被认为轻量是由于它运行于整个程序的上下文内,能使用整个程序共有的资源和程序环境。 作为单一顺序控制流,在运行的程序内线程必须拥有一些资源作为必要的开销。例如,必须有执行堆栈和程序计数器。在线程内执行的代码只在它的上下文中起作用,因此某些地方用"执行上下文"来代替"线程"。线程间的同步和通信1、互斥锁mutex——unlocklock用于线程需读/写一个共享数据段时。2、条件变量——防止互斥访问时发生死锁。3、信号量机制1私用信号量private samephore——实现同一进程中各线程之间的同步。2公用信号量public samephore——实现不同进程间或不同进程中各线程之间的同步。

三.我认为进程是一个正在执行的程序，可以分配给处理器并由处理器执行的一个实体，由单一顺序的执行显示，一个当前状态和一组相关的系统资源所描述的活动单元、进程的三种状态分别为就绪状态、执行状态和阻塞状态。进程的调度无非就是进程的三种状态的相互转换过程，进程的就绪、执行、阻塞、相互转换以及进程管理过程中应用到的几种原语就绪Ready、执行Run、阻塞Block、进程的申请P、归还进程的操作V，还有进程实体其的特征：1) 动态性——由创建而产生，由调度而执行，由撤销而消亡。2) 并发性——多个进程实体同存与内存中，且能在一段时间内同时运行。3) 独立性——独立运行、分配资源和接受调度。4异步性——进程按各自独立的不可预知的速度向前推进。 进程控制块PCB——OS是根据PCB来对并发执行的进程进行控制和管理的信息1) 进程标识符——惟一的标志一个进程。有内部系统使用和外部用户使用两种标识符。2) 处理机状态——信息由处理机的各种寄存器中的内容组成。处理机在运行时信息存储在寄存器中在中断时存在PCB中支持断点执行。寄存器a.通用寄存器用户可视寄存器——用户访问暂存信息。b.指令计数器——存放了要访问的下一条指令的地址。c.程序状态字PSW——状态信息，如条件码、执行方式、中断屏蔽标志等d.用户栈指针——存放过程和系统调用参数及调用地址，栈指针指向栈顶。3) 进程调度信息——a.进程状态、b.进程优先级、c.与进程调度算法有关的信息、d.事件4）进程控制信息——a.程序和数据的地址、b.进程同步和通信机制、c.资源清单、d.链接指针进程控制块的组织方式1)链接方式2)索引方式，以上这些是我对于进程管理这一部分的相关理解。

四课程设计过程中对相应部分的设计与实现思路

课程设计中运用了链表的增删改，同时运用了位视图可以非常方便的找到一颗应用的pcb相应的流程图如下：通过程序的帮助文档选择需要的操作可以满足用户进行创建进程、阻塞进程、唤醒、以及时间片到各个功能的具体转换如下：阻塞、执行——阻塞、唤醒、阻塞------就绪、时间片到、就绪——执行、执行--------就绪。程序的设计思路就是根据进程的调度及相互转换的过程，利用一些链表、函数等数据结构完成的。

五读书工程心得总结

通过此次读书工程，查找资料并结合课程设计，使我深刻感受到了编程的重要性，之前我编程的能力非常弱也对编程很不感兴趣，因为自己以后并不打算以计算机为职，所以对于编程这个弱项一直都不怎么在意认为自己以后也用不到，但从大一以来的各种实验课都上的十分痛苦，每当查实验的时候更是纠结烦躁，因为非常害怕几乎都不笑的金老师，才让我慢慢学着编写程序不断的调试代码。

操作系统这门课已经结束了，通过这个学期的的学习，让我了解了计算机内部的很多功能的实现方法以及结构原理。这次读书报告让我深刻了解了进程这一概念，加深了我对进程之间的转换、进程的状态、以及进程的更多了解。使我了解了有关操作系统的知识以及当前现状下操作系统的发展前景。并且读了其他的书籍也让我了解了课本上没有的东西。正常教学周学习的课本上让我知道了PCB这个东西没进程控制块，了解了进程的调度三种基本状态及相互转换的过程，并了解关于进程、线程方面的知识，并能知道两者之间的区别和联系。通过编程的过程让我对链表和结构体有了更进一步的了解并能运用得更加熟练。通过本学期的上机实验让我在编程上的思路渐渐清晰有不同的想法。

参考文献：操作系统设计与实现第二版有电子书中文版作者(美)Andrew S. Tanenbaum,Albert S. Woodhull译者：陈渝、谌卫军，出版社：电子工业出版社，出版日期：2007年3月； 操作系统概念精要(Operating System Concepts Essentials)(2011年版)(英文版有电子书)作者ABRAHAM SILBERSCHATZ、PETER BAER GALVIN、GREG GAGNE 出版社：JOHN WILEY & SONS. INC，出版日期：2011年 ；Windows操作系统原理第2版作者：陈向群、向勇、王雷、马洪兵、郑扣根、美DavaProbert，出版社：机械工业出版社，出版日期：2004年11月；