**操作系统课程设计**

**总结报告**

**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**

|  |
| --- |
| 本学期开设了操作系统课程设计。   1. 进程控制   1.1目的  利用简单的结构和控制方法模拟进程结构、进程状态和进程控制。  1.2内容  用PCB表示整个进程实体，利用随机数方法或键盘控制方法模拟进程执行中产生的事件。或者利用鼠标或者键盘中断的基于图形接口方式的进程控制管理。  具体过程：  1、 定义PCB（可以采用静态结构或动态结构）：包括理论PCB中的基本内容，如内部ID、外部ID、进程状态、队列指针。由于无法实现真正的进程创建功能，在实验中只需建立PCB，用它代表完整的进程。  2、 定义进程状态转换方式：进程的状态转换是由进程内部操作或操作系统的控制引起，由于无法实现这些功能，学生可以采用随机数方法或键盘控制方法模拟，并实现对应的控制程序。随机方法指产生1－6的随机数，分别代表创建进程(c)、结束进程(e)、进程阻塞(b)、激活进程(w)、调度进程(p)、时间片到(t)等事件；键盘模拟方法指定义6种按键代表以上6种事件。  3、 根据四种事件处理就绪队列、阻塞队列和当前执行中的进程。  4、 每次事件处理后应形象地显示出当前系统中的执行进程是哪一个，就绪队列和阻塞队列分别包含哪些进程。  1.3数据结构  创建就绪、阻塞、执行链表，在就绪中申请进程结点，若执行为空则直接删除就绪中结点添加进执行中。阻塞进程时把正在执行的进程删除，添加在阻塞链表中。激活后将阻塞链表中的相应结点删除添加到执行链表中。时间片到即将相应的执行结点删除放入就绪链表中。  1.4算法设计及流程图  执行状态  就绪状态  键入名称  阻塞状态  是否重名  时间片是否足够  执行状态是否为空  开始创建  结束  唤醒  1.5小结  进程控制是进程管理中最基本的功能。它作用于创建一个新进程，终止一个已完成的进程，或终止一个因出现某事件而无法运行下去的进程，还可以负责进程运行中的状态转换。本实验就是从进程的创建以及各种状态出发进行虚拟模拟，通过代码分析与编写，我已经掌握了进程控制的基本理论与实现。  二、请求分页存储区管理  2.1目的  实现分页式存储地址转换过程，在此基础上实现请求分页的地址转换。分页式存储管理系统是内存非连续存储管理中基本的方法，可以通过把一个作业分成多个页面分配到不连续的内存块中去。实验可以通过位示图的方式来模拟内存的使用情况，为每个作业建立页表用于完成正确的地址转换。  2.2内容  利用键盘输入本模拟系统的物理块的大小，作业的页表中的块号；完成逻辑地址转换成相应的物理地址的过程。  具体过程：   1. 建立一张位示图，用来模拟内存的分配情况，利用随机数产生一组0和1的数对应内存的使用情况。 2. 输入块（页）的大小，通过模拟位示图为本作业分配内存空间建立相应的页表（长度不定）； 3. 录入逻辑地址转换成相应的物理地址 4. 扩充页表，变成请求式的二维页表（增加存在位等）完成地址转换。   2.3数据结构  首先通过位示图判断页数。请求分页存储区管理算法分为先进先出算法和最近最久未使用算法，通过公式计算从物理地址到逻辑地址的转换，累加置换次数。缺页次数。用压栈出栈的方式实现编程。  2.4算法设计及流程图    2.5小结  在进程运行过程中，若其所要访问的页面不在内存而需要把它们调入内存，但内存已无空闲空间时，为了保证该进程能正常运行，系统必须从内存中调出一页程序或者数据送磁盘的对换区中。先进先出置换算法和最近最久未使用置换算法是常见的置换算法。先进先出置换算法总是淘汰最先进入内存的页面，只需把一个进程已调入内存的页面，按先后次序连接成一个队列，并设置一个指针，使它总是指向最老的页面。最近最久未使用的页面置换算法，是根据页面调入内存后的使用情况进行决策的。由于无法预测个页面将来使用情况，只能利用最近的过去作为最近的将来的近似。LRU置换算法是选择最近最久未使用的页面予以淘汰。  三、设备管理  3.1目的  设备管理主要包括设备的分配和回收、同时实现设备独立性。  3.2内容  假定模拟系统中有A（3台）、B（2台）、C三种设备，采用安全分配方式。  1、设备管理子系统涉及到系统设备表（SDT）、通道控制表（CHCT）、控制器控制表（COCT）和设备控制表（DCT）来体现输入输出系统的四级结构和三级控制。我们模拟这样的数据结构来完成对外围设备的管理。  （1）添加设备：增加对应的设备控制表和系统设备表中的表项，如果需要新建对应的控制器控制表。  （2）删除设备：删除对应的设备控制表和系统设备表中的表项，如果需要删除对应的控制器控制表。  2、设备分配必须满足设备的独立性要求。为了实现设备独立性，要求在驱动程序之上设计一层设备无关软件，其主要功能可分为：  （1）执行所有设备的公有操作，主要包括：(a)独占设备的分配与回收；(b)将逻辑设备名映射为物理设备（LUT），进一步可以找到相应物理设备的驱动程序。  （2）向用户层（或文件层）软件提供统一的接口。例如，对各种设备的读操作，在应用程序中都用read; 而对各种设备的写操作，则都使用write。  3.3数据结构  ＣＨＣＴ　ＣＨ１――ｃｏ１　Ｋ    　　　　　　　　　　　　　　Ｍ  　　　　　　　　　ｃｏ２―――Ｐ  　　　　　ＣＨ２  　　　　　　　　　ｃｏ３―――Ｔ  　　根据各控制器之间的关系，形成链表  3.4算法设计及流程图  申请1个新的COCT  申请1个新的DCT  挂起在设备控制器等待队列  挂起在通道等待队列  设备分配成功  进程申请设备  挂起在设备等待队列  查看SDT中是否有申请的设备  当前状态是否忙  查看COCT中是否有申请的设备控制器  当前状态是否忙  查看CHCT中申请的设备是否忙  否  是  是  否  否  是  是  否  是  否  3.5小结  在多道程序环境下，系统中的设备供所有进程共享。为防止诸进程对系统资源的无序竞争，条规定系统设备在不允许用户自行使用，必须有系统统一分配。每当进程向系统分配给请求用户。在有的系统中，为了确保在ＣＰＵ与设备之间能进行通信，还应分配相应的控制器和通道。为了实现设备分配，必须在系统中设置相应的数据结构。  四、文件管理  4.1目的  利用交互式命令实现树型目录结构和文件管理。  4.2内容  在文件中保存目录内容，创建文件或子目录可以用命令行命令：MD、CD、RD、MK（创建文件）、DEL（删除文件）和DIR。目录项包括文件或目录名称、类型（文件、目录或空目录项）、创建日期以及下一个目录项指针、下一级目录项指针。  具体过程：  1、创建初始文件，建立根目录的“.”和“..”目录项。  2、显示命令提示符“$”。  3、输入命令后根据命令含义完成相应文件操作：   * + MD：在目录文件中创建子目录，同时搜索当前目录最后一个目录项，并保存指针信息；   + CD：根据当前目录切换到指定目录；   + RD：搜索所要删除的目录是否为空目录，若是则删除；   + MK：在当前目录中创建文件名称；   + DEL：搜索所要删除的文件是否存在，若是则删除；   + DIR：列出当前目录的所有目录项。   4.3数据结构  创建目录时，在目录文件中创建子目录，同时搜索当前目录最后一个目录项，并保存指针信息。删除目录，搜索所要删除的目录是否为空目录，若是则删除。创建文件，即进入目录后，在当前目录中创建文件名称。删除文件，搜索所要删除的文件是否存在，若是则删除。查看目录，列出当前目录的所有目录项。  4.4算法设计及流程图  选择操作  创建文件夹  删除文件夹  进入文件夹  创建文件  删除文件  4.5小结  在现代ＯＳ中，几乎毫无例外的是通过文件系统来组织和管理在计算机中所存储的大量程序和数据的，文件系统的功能是通过把它所管理的程序和数据组织成一系列文件的方法来实现的。通常，为了能对这些文件实施有效的管理，必须对他们加以妥善组织。文件也是一种数据结构，用于表示系统中的文件及物理地址，供检索使用。多级树形目录算法可以实现创建目录，删除目录，在目录下创建文件和删除文件。  五、银行家算法  5.1目的  很行家算法是具有代表性的避免死锁的算法，是Dijkstra的银行家算法。本次实验用来帮助学生深刻体会银行家算法的实施之处。  5.2内容  每次申请资源时要先判断当前时刻是否存在安全序列，并且如果分配资源是否还存在安全序列。  5.3数据结构  typedef struct Process //--进程  {  string name; //--进程名称  int Max[3]; //--进程最大资源  int Allocation[3]; //--进程已有资源  int state; //--状态位  int Need[3]; //--进程还需资源  struct Process \*next;  }\*process;  5.4算法设计及流程图    5.5小结  本实验主要是银行家算法的编程，进一步通过实践的 操作加深了对银行家算法的 理解和掌握。但在实现的过程中遇到了很多问题，如：算法的具体实现和程序的调试等。我用的是C语言编写的，在最后的安全性算法中遇到一个问题：执行结果总是出错，后来经过反复的思考和调试，才发现原来是由循环里面的count0值未初始化为0所导致，后经修改得到正确的结果。  六、总结  通过一学期的操作系统的学习与实验的实践，我从完全不会编程到对变成产生了一些兴趣，这是我本学期最大的进步。虽然有些实验还需要经过其他同学的帮助，但在每一个实验中我都有所收获。  本学期的五个实验：进程控制，请求分页存储管理，设备管理，文件管理，银行家算法。我都是首先从理论出发，层层深入的理解其中的原理，透彻分析后才下手编程。因为基础比较差，用的时间也比较多，有些实验未能及时完成。但最终还是在老师和同学的帮助下成功了。本学期是我在大学编程进步最大的一个学期，以后会继续努力。 |