### 实验一、进程调度

1. **实验目的**

用高级语言编写和调试一个进程调度 程序，以加深对进程的概念及进程调度算法的理解。

**二、实验内容和要求**

编写并调试一个模拟的进程调度程序，采用“简单时间片轮转法”调度算法对五个进程进行调度。

每个进程有一个进程控制块（ PCB）表示。进程控制块可以包含如下信息：进程名、到达时间、需要运行时间、已运行时间、进程状态等等。

进程的到达时间及需要的运行时间可以事先人为地指定（也可以由随机数产生）。进程的到达时间为进程输入的时间。 进程的运行时间以时间片为单位进行计算。

每个进程的状态可以是就绪 W（Wait）、运行R（Run）两种状态之一。

就绪进程获得 CPU后都只能运行一个时间片。用运行时间加1来表示。

如果运行一个时间片后，进程的已占用 CPU时间已达到所需要的运行时间，则撤消该进程，如果运行一个时间片后进程的已占用CPU时间还未达所需要的运行时间，也就是进程还需要继续运行，此时应分配时间片给就绪队列中排在该进程之后的进程，并将它插入就绪队列队尾。 每进行一次调度程序都打印一次运行进程、就绪队列、以及各个进程的 PCB，以便进行检查。

重复以上过程，直到所要进程都完成为止。

**三、实验主要仪器设备和材料**

硬件环境：IBM-PC或兼容机

软件环境：C语言编程环境

**四、实验原理及设计方案**

1、进程调度算法：采用多级反馈队列调度算法。其基本思想是：当一个新进程进入内在后，首先将它放入第一个队列的末尾，按FCFS原则排队等待高度。当轮到该进程执行时，如能在该时间片内完成，便可准备撤离系统；如果它在一个时间片结束时尚为完成，调度程序便将该进程转入第二队列的末尾，再同样地按FCFS原则等待调度执行，以此类推。

2、实验步骤:

（1）按先来先服务算法将进程排成就绪队列。

（2）检查所有队列是否为空，若空则退出，否则将队首进程调入执行。

（3）检查该运行进程是否运行完毕，若运行完毕，则撤消进程，否则，将该进程插入到下一个逻辑队列的队尾。

（4）是否再插入新的进程,若是则把它放到第一逻辑队列的列尾。

（5）重复步骤（2）、（3）、（4），直到就绪队列为空。

**3 流程图：**

进程完成，撤消该进程

就绪队列首进程投入运行

时间片到，运行进程已占用CPU时间+1

运行进程已占用CPU时间已达到所需的运行时间

把运行进程插入到下一个队列的队尾

插入新的进程

开始

初始化PCB,输入进程信息

所有队列都为空

各进程按FCFS原则排队等待调度

退出程序

是

**是**

**五、结果过程及截图**

**六、所遇困难的解决以及心得体会**

**七、源代码**

**实验二、进程同步与互斥之睡眠理发师问题**

**【实验目的】**

利用信号量和PV操作来实现进程同步与互斥，让学生掌握进程同步与互斥的原理，以及解决进程同步与互斥的算法，从而进一步巩固进程同步与互斥等相关的内容。

**【实验内容和要求】**

某一理发店有一名理发师、一把理发椅和n把供等候理发的顾客坐的椅子。

要求如下：

1. 理解并掌握信号量的概念；
2. 利用信号量解决睡眠理发师问题。

【**实验原理**】

某一理发店有一名理发师、一把理发椅和n把供等候理发的顾客坐的椅子。

1. 如果没有顾客，理发师便在理发椅上睡觉；
2. 一个顾客到来时，必须叫醒理发师
3. 如果理发师正在理发时，有顾客来到，则如果有空椅子可坐，就坐下来等待，否则就离开。

**实验三 主存空间的分配和回收**

**一、实验目的**

熟悉主存的分配与回收。理解在不同的存储管理方式下，如何实现主存空间的分配与回收。掌握动态分区分配方式中的数据结构和分配算法及动态分区存储管理方式及其实现过程。

**二、实验内容和要求**

主存的分配和回收的实现是与主存储器的管理方式有关的。所谓分配，就是解决多道作业或多进程如何共享主存空间的问题。所谓回收，就是当作业运行完成时将作业或进程所占的主存空间归还给系统。

可变分区管理是指在处理作业过程中建立分区，使分区大小正好适合作业的需求，并且分区个数是可以调整的。当要装入一个作业时，根据作业需要的主存量查看是否有足够的空闲空间，若有，则按需要量分割一个分区分配给该作业；若无，则作业不能装入，作业等待。随着作业的装入、完成，主存空间被分成许多大大小小的分区，有的分区被作业占用，而有的分区是空闲的。

实验要求使用可变分区存储管理方式，分区分配中所用的数据结构采用空闲分区表和空闲分区链来进行，分区分配中所用的算法采用首次适应算法、循环首次适应算法、最佳适应算法三种算法来实现主存的分配与回收。同时，要求设计一个实用友好的用户界面，并显示分配与回收的过程。

**三、实验主要仪器设备和材料**

硬件环境：IBM-PC或兼容机

软件环境：VC++ 6.0

**四、实验原理及设计方案**

1、循环首次适应算法

在该算法中，把主存中所有空闲区按其物理地址递增的次序排列。在为作业分配存储空间时，从上次找到的空闲分区的下一个空闲分区开始查找，直到找到第一个能满足要求的空闲区，从中划出与请求的大小相等的存储空间分配给作业，余下的空闲区仍留在空闲区表或链中。

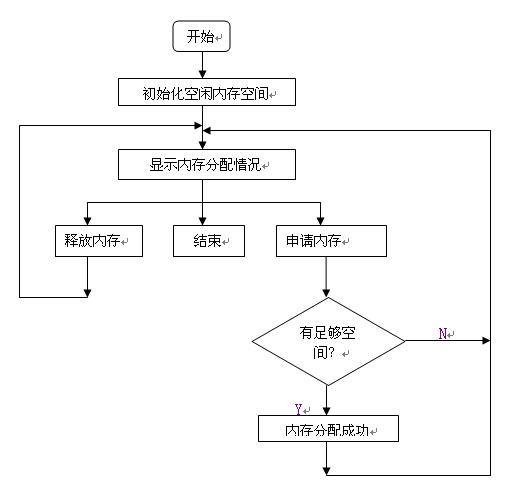
1. 实验步骤

（1）初始化空闲分区；

（2）反复对现有的空闲分区进行进程创建和撤消，即内存分配和回收；

（3）退出。

3、流程图



输入要释放内存的基地址和大小

**五、结果过程及截图**

**六、所遇困难的解决以及心得体会**

**七、源代码**

**实验四 文件系统**

**一、实验目的**

模拟文件系统实现的基本功能,了解文件系统的基本结构和文件的各种管理方法,加深理解文件系统的内部功能及内部实现。通过用高级语言编写和调试一个简单的文件系统，模拟文件管理的工作过程，从而对各种文件操作命令的实质内容和执行过程有比较深入的了解。

**二、实验内容和要求**

编程模拟一个简单的文件系统，实现文件系统的管理和控制功能。要求本文件系统采用两级目录，即设置主文件目录[MFD]和用户文件目录[UED]。另外，为打开文件设置运行文件目录[AFD]。设计一个10个用户的文件系统，每次用户可保存10个文件，一次运行用户可以打开5个文件，并对文件必须设置保护措施。在用户程序中通过使用文件系统提供的Create、open、read、write、close、delete等文件命令，对文件进行操作

**三、实验主要仪器设备和材料**

硬件环境：IBM-PC或兼容机

软件环境：C语言编程环境

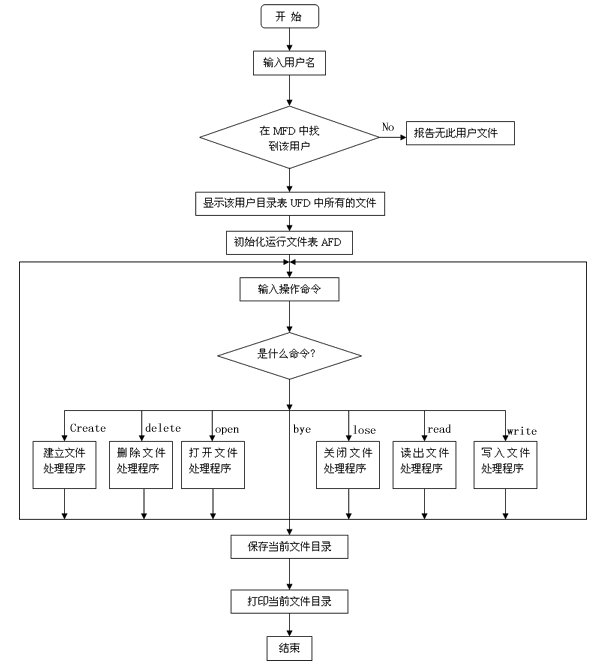
**四、实验原理及设计方案**

1、实验原理

运用二级目录思想来模拟文件系统。

为每个用户建立一个单独的用户文件目录UFD。这些文件目录具有相似的结构，它由用户文件的文件块组成。此外，在系统再建立一个主文件目录MFD；在主文件目录中，每个用户目录都占有一个目录项，其目录项中包含文件名和指向该文件目录文件的指针。

3、程序流程图

****

**五、结果过程及截图**

**六、所遇困难的解决以及心得体会**

**七、源代码**

**实验五 磁盘调度算法**

**【实验目的】**

通过实验使学生了解并掌握最短寻道时间优先算法的相关特性。

**【实验要求】**

实验结束后要求每位同学都应该掌握最短寻道时间优先算法的基本原理。

【**实验内容及原理**】

(1)给定一组测试数据，如：20，44，40，4，80，12，76；读写头起始位置：40。创建文件job.txt，并导入测试数据：

40 7

98，181，35，112，24，144，45，57

注：42表示起始位置，7表示要访问的磁道数。

(2)编译源代码

(3)观察并结合运行结果研究实例代码中最短寻道时间优先算法的特点，补充先来先服务算法的代码，运行并分析比较这两种算法的优缺点及区别。

【**算法描述**】

(1)最短寻道时间优先算法

这种算法的基本出发点是以磁头移动距离的大小作为优先的因素。它从当前磁头位置出发，选择离磁头最近的磁道为其服务。最短寻道时间优先算法使那些靠近磁头当前位置的申请可及时得到服务，防止了磁头大幅度来回摆动，减少了磁道平均查找时间。该算法没考虑磁头移动的方向，也没有考虑进程在队列中等待的时间，从而可能使移动臂不断花时间改变方向，还可能使一些离磁头较远的申请者在较长时间内得不到服务。

(2)先来先服务算法

先来先服务算法根据访问请求的先后次序选择先提出访问请求的为之服务。

先来先服务算法是磁盘调度的最简单的一种形式，它既容易实现，又公平合理。它的缺点是效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，致使磁头反复移动，增加了服务时间，对机械的寿命也不利。

(3)循环扫描算法

循环扫描算法是对扫描算法的改进。如果对磁道的访问请求时均匀分布的，当磁头到达磁盘的一端并反向运动时，落在磁头之后的访问请求相对较少。这是由于这些磁道刚被处理，而磁盘另一端的请求密度相当高，且这些访问请求等待的时间长。为了解决这种情况,循环扫描算法规定磁头单向移动。例如,只自里向外移动,当磁头移到最外的被访问磁道时,头立即返回到最里的欲访问磁道,即将最小磁道号紧接着最大磁道号构成循环,进行扫描。

**实验六 进程同步与互斥之司机-售票员问题**

**书91页**