**附录 实验报告**

**课程编号：A0801040060**

**操作系统**

**实验报告**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | **徐婧** | **班级学号** | **软件2002 20206647** | **成绩** |  |
| **报告名称** | **操作系统实验报告** | | | | |
| **开设学期** | **2021-2022春季学期** | | | | |
| **开设时间** | **第16周——第17周** | | | | |
| **报告完成日期** |  | | | | |
| **批改教师** |  | | | | |
| **批改日期** | **2022年6月20** | | | | |

**东北大学软件学院**

**2022年6月**实验名称（例如：实验一-进程/线程的同步与互斥）

**一、实验目的**

1. 加深对进程概念/线程概念的理解；
2. 进一步认识并发的实质；
3. 分析线程、进程竞争资源的现象，学习解决进程互斥的方法；
4. 了解Windows对进程/线程管理的支持。

**二、问题分析与程序设计**

**（一）问题分析：**

面对该问题，必须让生产者在缓冲区满时休眠（或直接放弃数据），等到下次消费者消耗缓冲区中的数据的时候，生产者才能被唤醒，开始往缓冲区添加数据。也可以让消费者在缓冲区空时进入休眠，等到生产者往缓冲区添加数据之后，再唤醒消费者。

通常采用进程间通信的方法解决该问题。如果解决方法不完善，会容易出现死锁的情况。出现死锁时，两个线程都会陷入休眠，等待对方唤醒自己。

注意：在缓冲区为空的时候，消费者不能再进行消费；缓冲区为满时，生产者不能再生产。在一个线程进行生产或消费时，其余进程不能再进行此操作。

（二）**程序设计：**

1. 通过继承Runable接口并实现他的run方法来创建线程；
2. 定义一个容器类，该容器类中具有push方法和pop方法，分别对应往容器中放产品和取产品，由于在同一时间段只能有一个线程访问该该方法，所以给这两个方法加锁；
3. 分别定义生产者类和消费者类，他们均是不同的线程；
4. 定义测试类；
5. **实现过程与测试结果分析**

1.缓冲区类：

**生产者方法：**



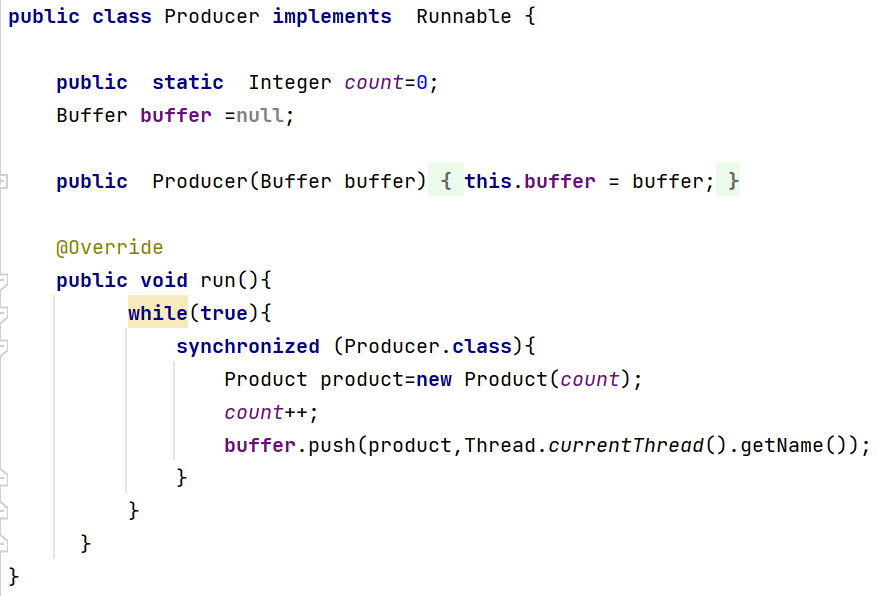
**消费者方法：**

****

push、pop方法生产和清理产品，由于同一时间段只能有生产或消费一个进程访问该方法，所以要给这两个方法加锁，使用wait()/notify()方法，当缓冲区已满时，生产者线程停止执行，放弃锁，使自己处于等状态，让其他线程执行；当缓冲区已空时，消费者线程停止执行，放弃锁，使自己处于等状态，让其他线程执行；

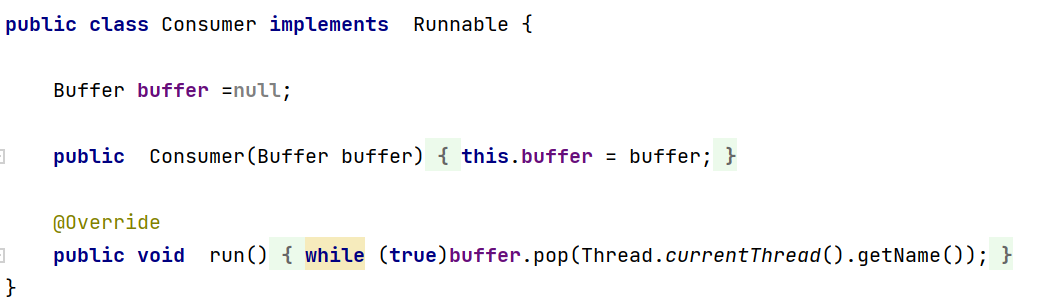
当生产者向缓冲区放入一个产品时,向其他等待的线程发出可执行的通知,同时放弃锁，使自己处于等待状态;当消费者从缓冲区取出一个产品时,向其他等待的线程发出可执行的通知,同时放弃锁,使自己处于等待状态。

**2.生产者类：**

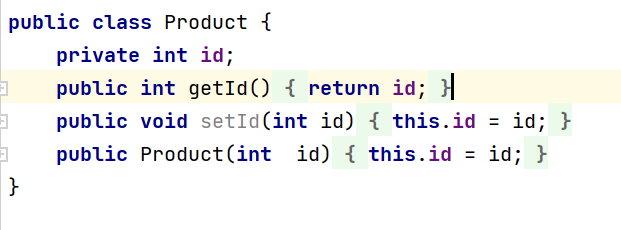


通过继承Runable接口并实现他的run方法来创建生产线程。

**3.消费者类：**

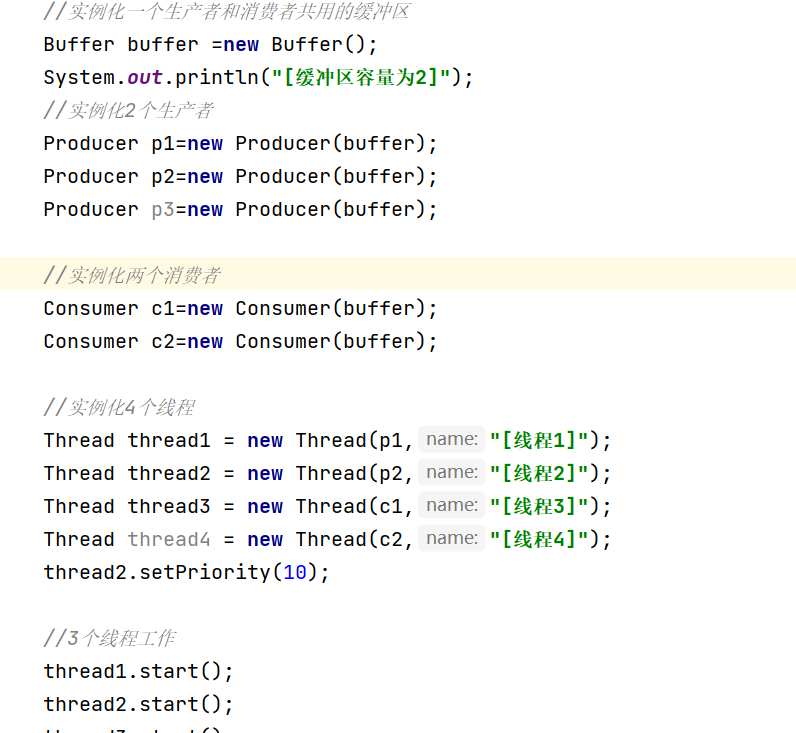
****

**4.产品类：**

****

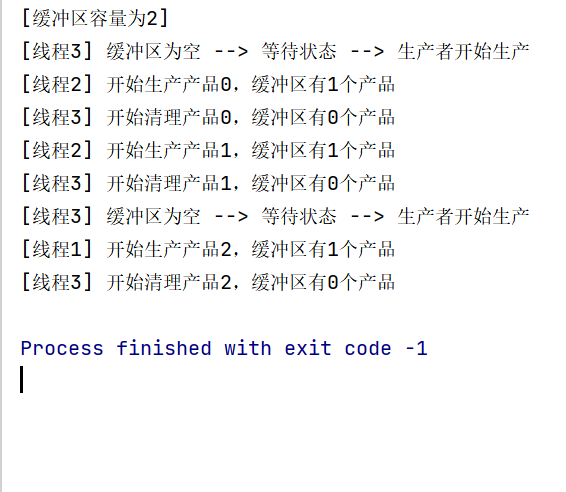
定义产品Id。

**5.测试类**



实例化生产者、消费者、线程。

**四、实验结果总结**



**五、创新的部分**

1. 使用wait()/notify()方法，给push和pop两个方法加锁；
2. 使用线程的睡眠Thread.sleep()方法。

**六、对实验的意见与建议**

通过该次实验，我加深对进程概念和线程概念的理解；进一步认识并发实质，分析进程以及线程竞争资源的现象，学习解决进程互斥的方法。

**七、思考题**

1. **如何控制进程间的相互通信？**

1）管道：管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

2）消息队列：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

3）信号量：信号量是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。

4）共享内存：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号两，配合使用，来实现进程间的同步和通信。

5）FIFO：FIFO可以在无关的进程之间交换数据，与无名管道不同；FIFO有路径名与之相关联，它一一种特殊设备文件形式存在于文件系统中。

1. **什么是进程同步？什么是进程互斥？分别有哪些实现方式？**

进程同步是指在多道程序环境下，进程是并发执行的，不同进程之间存在着不同的相互制约关系。

**进程同步的实现方式：**

1）信号量：用于进程间传递信号的一个整数值。在信号量上只有三种操作可以进行：初始化，P操作和V操作，这三种操作都是原子操作。

2）管程：管程是由一个或多个过程、一个初始化序列和局部数据组成的软件模块。

3）消息传递：消息传递的实际功能以一对原语的形式提供：send(destination,message)、

receive(source,message)

**进程互斥**是指两个或两个以上的进程，不能同时进入关于同一组共享变量的临界区域，否则可能发生与时间有关的错误，这种现象被称作进程互斥，也就是说，一个进程正在访问[临界资源](https://baike.baidu.com/item/%E4%B8%B4%E7%95%8C%E8%B5%84%E6%BA%90/1880269" \t "https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B%E4%BA%92%E6%96%A5/_blank)，另一个要访问该资源的进程必须等待。

**进程互斥的实现方式：**

1）关闭中断：使每个进程在进入临界区后先关闭中断，在离开之前再把中断打开。这样就能保证进程只有一个。

2）锁变量：设置一个共享的变量，初值为0，当进程想进入临界区时，会先测试锁；如果变量值为0，则进程将其置1并进入临界区，离开临界区后应该将锁变量的值置回0；若为1，则等待变量值变回0。

3）严格轮转法：严格轮换法同样也是针对一个临界区设置一个变量，假设为Turn。以两个进程为例子：当Turn为0时，Process 0才能能进入临界区，否则等待。等Process 0离开临界区后，将Turn设置为1.当Turn为1时，Process 1才能进入临界区，否则等待。等Process 1离开临界区后，将Turn设置为0。

实验二 处理机调度

**一、实验目的**

（1）加深对处理机调度的作用和工作原理的理解。

（2）进一步认识并发执行的实质。**二、问题分析与程序设计**

**第一题：设计一个按优先权调度算法实现处理器调度的程序。**

（1）假定系统有五个进程，每一个进程用一个进程控制块PCB来代表，进程控制块的格式为：进程名、指针、要求运行时间、优先数、状态。其中，

1）进程名——作为进程的标识，假设五个进程的进程名分别为P1，P2，P3，P4，P5。

2）指针——按优先数的大小把五个进程连成队列，用指针指出下一个进程的进程控制块的首地址，最后一个进程中的指针为“0”。

3）要求运行时间——假设进程需要运行的单位时间数。

4）优先数——赋予进程的优先权，调度时总是选取优先数大的进程先执行。

5）状态——可假设有三种状态，“就绪”状态（ready）、“运行”状态（working）和“结束”状态（finish）。五个进程的初始状态都为“就绪”，用“R”表示；当进程运行结束后，它的状态为“结束”，用“E”表示；当进程被选中开始运行但尚未结束时，它的状态为“运行”，用“W”表示。

（2） 在每次运行你所设计的处理器调度程序之前，为每个进程任意确定它的“优先数”和“要求运行时间”。

（3）为了调度方便，把五个进程按给定的优先数从大到小连成队列。用一单元指出队首进程，用指针指出队列的连接情况。

（4）处理器调度总是选队首进程运行。采用动态改变优先数的办法，进程每运行一次优先数就减“1”。由于本实验是模拟处理器调度，所以，对被选中的进程并不实际的启动运行，而是执行：

优先数-1

要求运行时间-1

来模拟进程的一次运行。

提醒注意的是：在实际的系统中，当一个进程被选中运行时，必须恢复进程的现场，让它占有处理器运行，直到出现等待事件或运行结束。在这里省去了这些工作。

（5）进程运行一次后，若要求运行时间≠0，则再将它加入队列（按优先数大小插入，且置队首标志）；若要求运行时间=0，则把它的状态修改成“结束”（E），且退出队列。

（6）若“就绪”状态的进程队列不为空，则重复上面（4）和（5）的步骤，直到所有进程都成为“结束”状态。

（7）在所设计的程序中应有显示或打印语句，能显示或打印每次被选中进程的进程名以及运行一次后进程队列的变化。

（8）为五个进程任意确定一组“优先数”和“要求运行时间”，启动所设计的处理器调度程序，显示或打印逐次被选中进程的进程名以及进程控制块的动态变化过程。

**第二题：设计一个按时间片轮转法实现处理器调度的程序。**

（1）假定系统有五个进程，每一个进程用一个进程控制块PCB来代表，进程控制块的格式为：进程名、指针、要求运行时间、已运行时间、状态。其中，

1）进程名——作为进程的标识，假设五个进程的进程名分别为P1，P2，P3，P4，P5。

2）指针——按优先数的大小把五个进程连成队列，用指针指出下一个进程的进程控制块的首地址，最后一个进程中的指针为“0”。

3）要求运行时间——假设进程需要运行的单位时间数。

4）已运行时间——假设进程已经运行的单位时间数，初始值为“0”。

5）状态——可假设有三种状态，“就绪”状态（ready）、“运行”状态（working）和“结束”状态（finish）。五个进程的初始状态都为“就绪”，用“R”表示；当进程运行结束后，它的状态为“结束”，用“E”表示；当进程被选中开始运行但尚未结束时，它的状态为“运行”，用“W”表示。

（2） 每次运行所设计的处理器调度程序前，为每个进程任意确定它的“要求运行时间”。

（3）把五个进程按顺序排成循环队列，用指针指出队列连接情况。另用一标志单元记 录轮到运行的进程。

（4）处理器调度总是选择标志单元指示的进程运行。由于本实验是模拟处理器调度的 功能，所以，对被选中的进程并不实际的启动运行，而是执行： 已运行时间+1 来模拟进程的一次运行，表示进程已经运行过一个单位的时间。

（5）进程运行一次后，应把该进程的进程控制块中的指针值送到标志单元，以指示下一个轮到运行的进程。同时，应判断该进程的要求运行时间与已运行时间，若该进程的要求运行时间¹已运行时间，则表示它尚未执行结束，应待到下一轮时再运行。若该进程的要求运行时间=已运行时间，则表示它已经执行结束，应指导它的状态修改成“结束”（E）且退出队列。此时，应把该进程的进程控制块中的指针值送到前面一个进程的指针位置。

（6）若“就绪”状态的进程队列不为空，则重复上面的（4）和（5）的步骤，直到所有的进程都成为“结束”状态。

（7）在所设计的程序中应有显示或打印语句，能显示或打印每次选中进程的进程名以及运行一次后进程队列的变化。

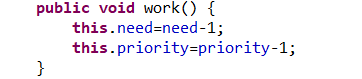
（8）为五个进程任意确定一组“要求运行时间”，启动所设计的处理器调度程序，显示或打印逐次被选中的进程名以及进程控制块的动态变化过程。输出要求与上述优先权优先调度输出形式类似。

**三、实现过程与测试结果分析**

**第一题：设计一个按优先权调度算法实现处理器调度的程序。**

每次循环找到最大优先数的进程，运行进程后要求运行时间-1，优先数-1；当所有进程的要求运行时间都到达0后，调度结束。

****

****

**第二题：设计一个按时间片轮转法实现处理器调度的程序。**

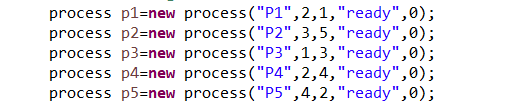
每次循环运行进程后，已运行时间+1；当所有进程的要求运行时间都等于已运行时间，调度结束。

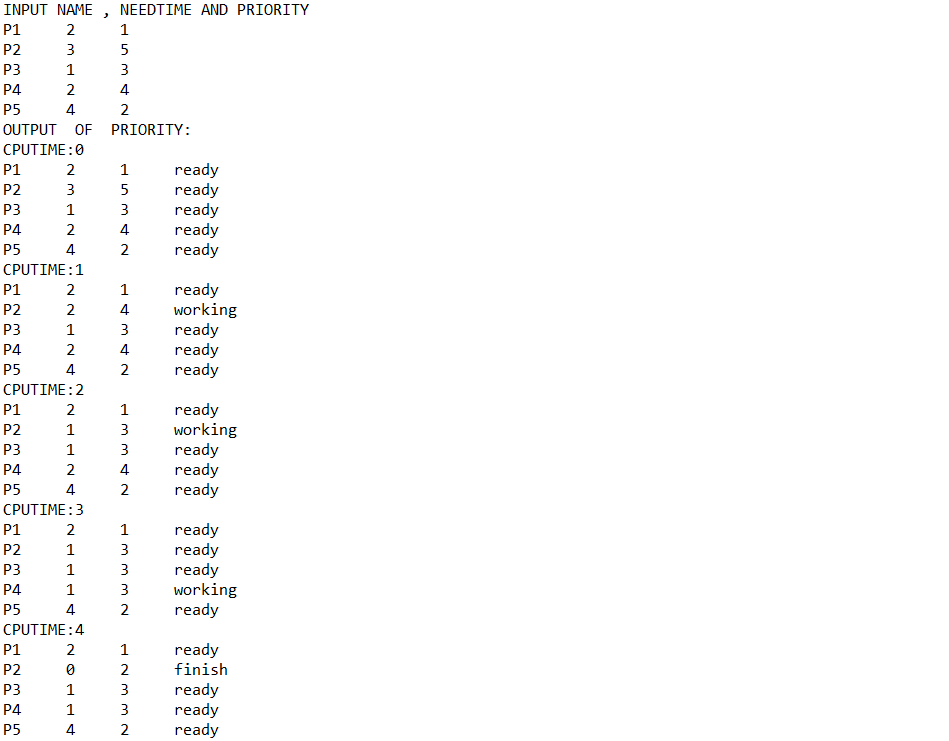


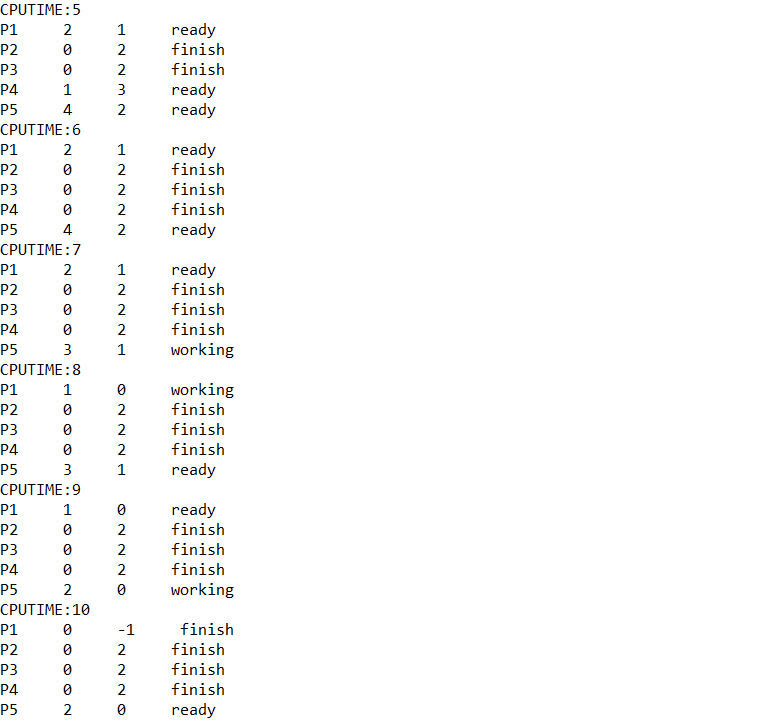


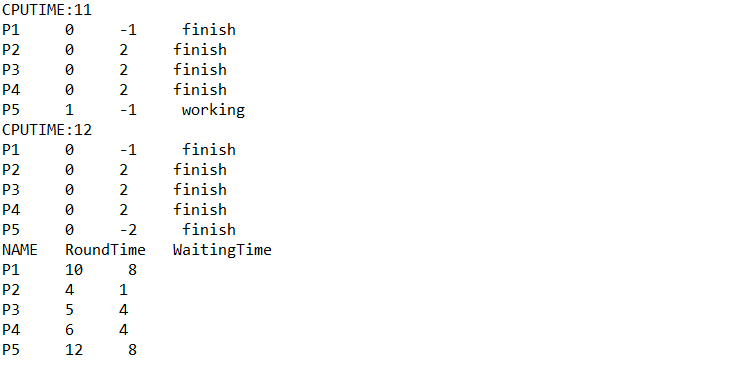
**四、实验结果总结**

**第一题：设计一个按优先权调度算法实现处理器调度的程序。**

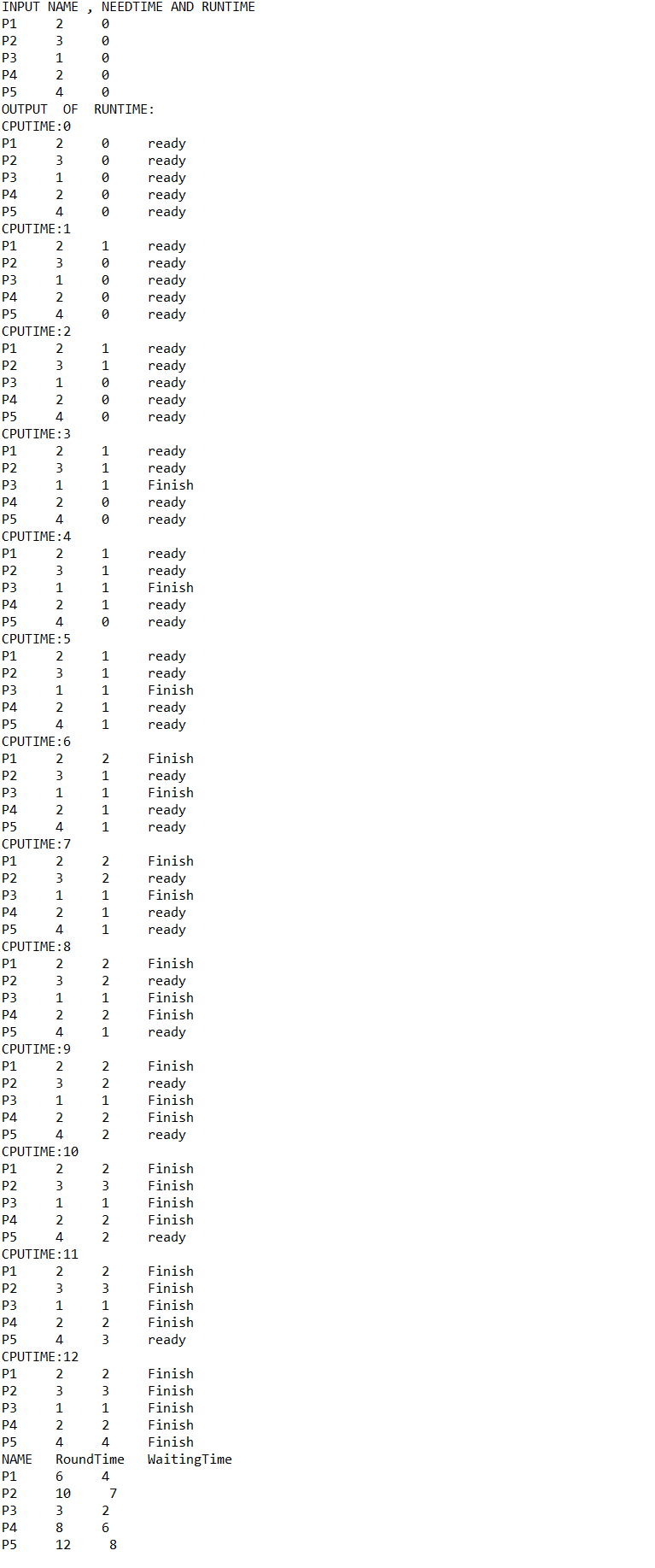


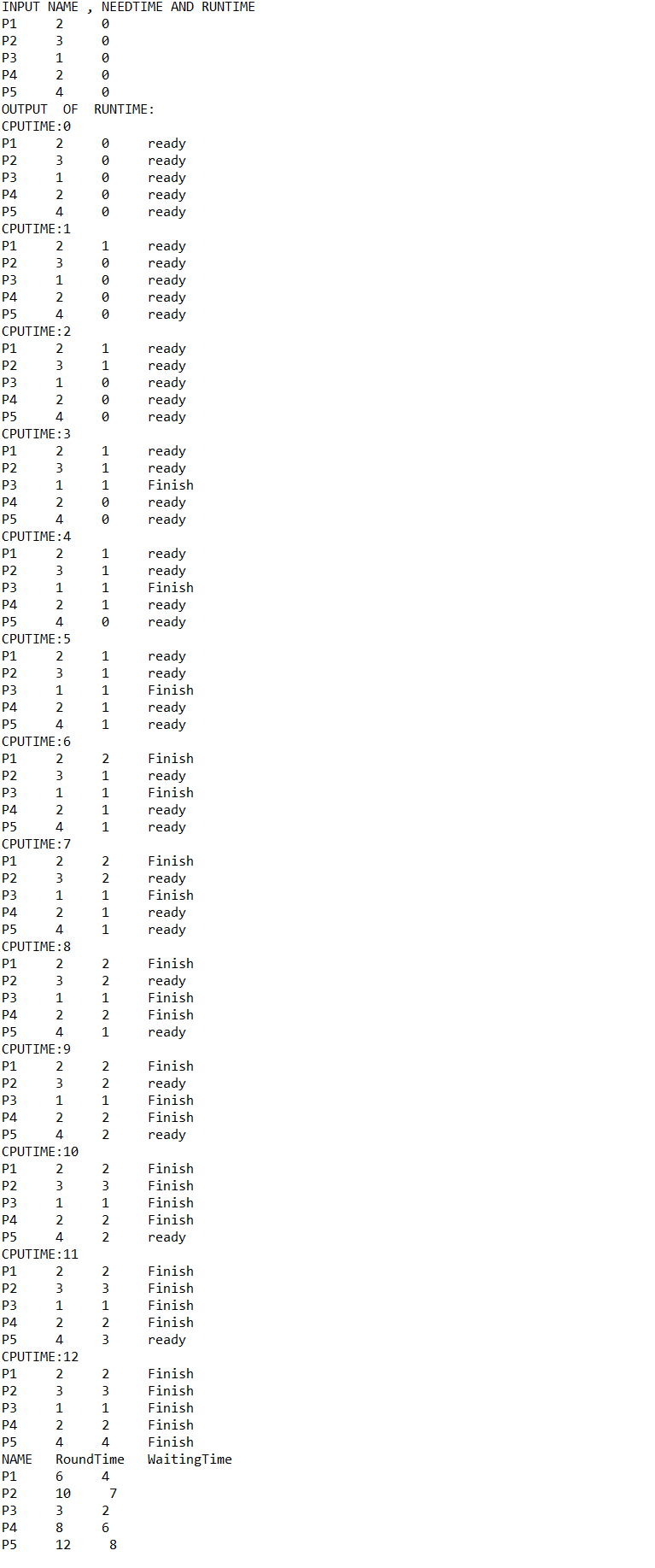






**第二题：设计一个按时间片轮转法实现处理器调度的程序。**





**五、创新的部分**

采用优先权调度的方法，实现处理器的调度。

采用时间片轮转法，实现处理器的调度。

**六、对实验的意见与建议**

通过本次实验，我更加深刻地理解了处理器优先权调度法和时间片轮转法，加深了我对处理器的理解，并且能够更加实际地体会将老师上课讲述的知识与实际相融合。

**七、思考题**

（1）处理机调度的目的？

合理快速的处理计算机软件硬件资源。

（2）你实现优先权调度算法的思想？

在每次对进程执行时找到优先数最高的可选进程，进程运行一次后，优先数-1，要求运行时间-1。若要求运行时间=0，则令进程退出队列。

（3）你采用时间片轮转法实现处理机调度的思想？

按队列顺序选择进程，将时间以一个cpu时间为单位，每次循环代表一次cpu时间，进程的已运行时间+1。进程运行一次后，应把该进程的进程控制块中的指针值送到标志单元，以指示下一个轮到运行的进程。判断该进程的要求运行时间与已运行时间，若尚未执行结束，应待到下一轮时再运行。若已经执行结束，应指导它退出队列。此时，应把该进程的进程控制块中的指针值送到前面一个进程的指针位置。

（4）比较效率如何？

优先权调度算法中，每次cpu时间执行完后都要寻找优先权最高的进程，浪费一定时间，但是每次都保证了执行当前优先级最高的进程。

实验三-存储管理

**一、实验目的**

1. 加深对存储管理的作用和工作原理的理解
2. 进一步认识主存空间的分配和处理方法
3. 进一步认识虚拟存储器的工作原理
4. 进一步认识文件系统的内部功能及内部实现
5. **问题分析与程序设计**

**第一题：模拟分页式存储管理中硬件的地址转换和产生缺页中断**

1. 分页式虚拟存储系统是把作业信息的副本存放在磁盘上，当作业被选中时，可把作业的开始几页先装入主存且启动执行。为此，在为作业建立页表时，应说明哪些页已在主存，哪些页尚未装入主存，页表的格式为：

|  |  |  |  |
| --- | --- | --- | --- |
| 页号 | 标志 | 主存块号 | 在磁盘上的位置 |
|  |  |  |  |

其中，

* 标志——用来表示对应页是否已经装入主存，标志位=1，则表示该页已经在主存，标志位=0，则表示该页尚未装入主存。
* 主存块号——用来表示已经装入主存的页所占的块号。
* 在磁盘上的位置——用来指出作业副本的每一页被存放在磁盘上的位置。

1. 作业执行时，指令中的逻辑地址指出了参加运算的操作数存放的页号和单元号，硬件的地址转换机构按页号查页表，若该页对应标志为“1”，则表示该页已在主存，这时根据关系式：

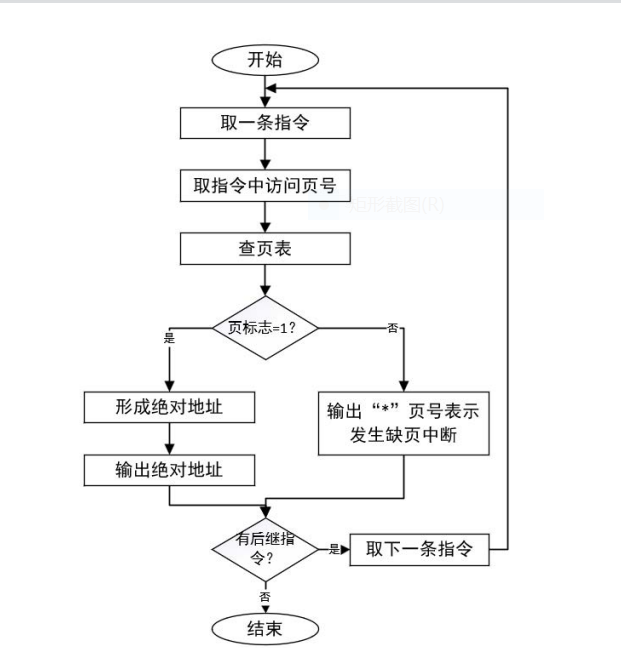
绝对地址=块号´块长+单元号

计算出欲访问的主存单元地址。若访问的页对应标志为“0”，则表示该页不在主存，这时硬件发“缺页中断”信号，由操作系统按该页在磁盘上的位置，把该页信息从磁盘读出装入主存后再重新执行这条指令。

1. 设计一个“地址转换”程序来模拟硬件的地址转换工作。当访问的页在主存时，

则形成绝对地址，但不去模拟指令的执行，而用输出转换后的地址来代替一条指令的执行。

当访问的页不在主存时，则输出“\*该页页号”，表示产生了一次缺页中断。算法如下：



**第二题：用先进先出（FIFO）页面调度算法处理缺页中断**

1. 在分页式虚拟存储系统中，当硬件发出“缺页中断”后，引出操作系统来处理这个中断事件。如果主存中已经没有空闲块，则可用FIFO页面调度算法把该作业中最先进入主存的一页调出，存放到磁盘上。然后再把当前要访问的页装入该块。调出和装入后都要修改页表中对应页的标志。

（2）FIFO页面调度算法总是淘汰该作业中最先进入主存的那一页，因此可以用一个数组来表示该作业已在主存的页面。假定作业被选中时，把开始的m个页面装入主存，则数组的元素可定为m个。例如：

P[0]，P[1]…，P[m-1]

其中，每一个P[i](I=0,1,…,m-1)表示一个在主存中的页面号。它们的初值为：

P[0]：=0,P[1]：=1,…,P[m-1]：=m-1

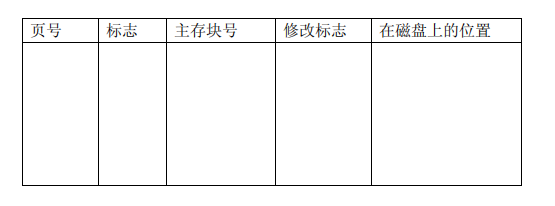
用一指针K指示当要装入新页时，应淘汰的页在数组中的位置，K的初值为“0”。当产生缺页中断后，操作系统选择 P[k]所指出的页面调出，然后执行：

P[k]：=要装入页的页号

k：=(k+1) mod m

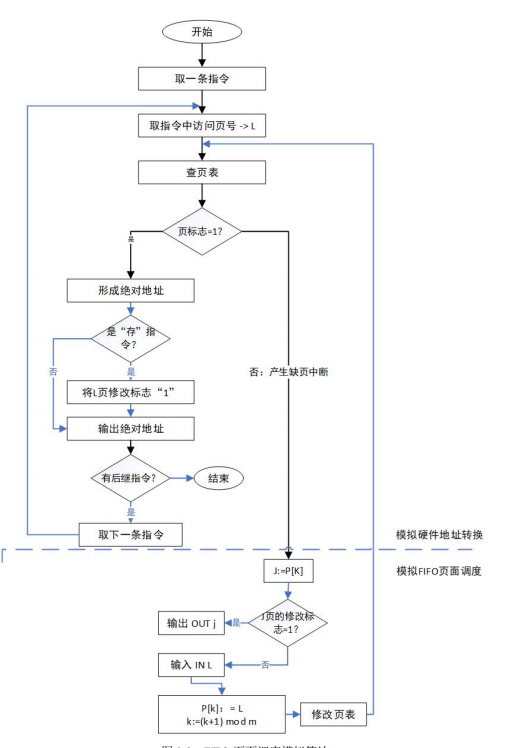
再由装入程序把要访问的一页信息装入到主存中。重新启动刚才那条指令执行。

（3）编制一个FIFO页面调度程序，为了提高系统效率，如果应淘汰的页在执行中没有修改过，则可不必把该页调出（因在磁盘上已有副本）而直接装入一个新页将其覆盖。因此在页表中增加是否修改过的标志，为“1”表示修改过，为“0”表示未修改过，格式为：



由于是模拟调度算法，所以，不实际地启动调出一页和装入一页的程序，而用输出调出的页号和装入的页号来代替一次调出和装入的过程。

把第一题中程序稍作改动，与本题结合起来，FIFO 页面调度模拟算法如图。



（4）如果一个作业的副本已在磁盘上，在磁盘上的存放地址以及已装入主存的页和作业依次执行的指令序列都同第一题中所示。于是增加了“修改标志”后的初始页表为：



按依次执行的指令序列，运行你所设计的程序，显示或打印每次调出和装入的页号，以及执行了最后一条指令后的数组P的值。

1. 为了检查程序的正确性，可再任意确定一组指令序列，运行设计的程序，核对执行的结果。

**第四题：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**

设计一个n个用户的简单二级文件系统，每次用户可保存m个文件，用户在一次运行中只能打开一个文件，对文件必须设置保护措施。

1.可以实现下列几条命令（至少4条）：login 用户登录；dir列文件目录；create创建文件；delete删除文件；open打开文件；close关闭文件；read读文件；write写文件。

2.列目录时要列出文件名、物理地址、保护码和文件长度。

3.源文件可以进行读写保护。

（1）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。

（2）程序采用二级文件目录（即设置主目录[MFD]）和用户文件目录（UED）。另外，为打开文件设置了运行文件目录（AFD）。

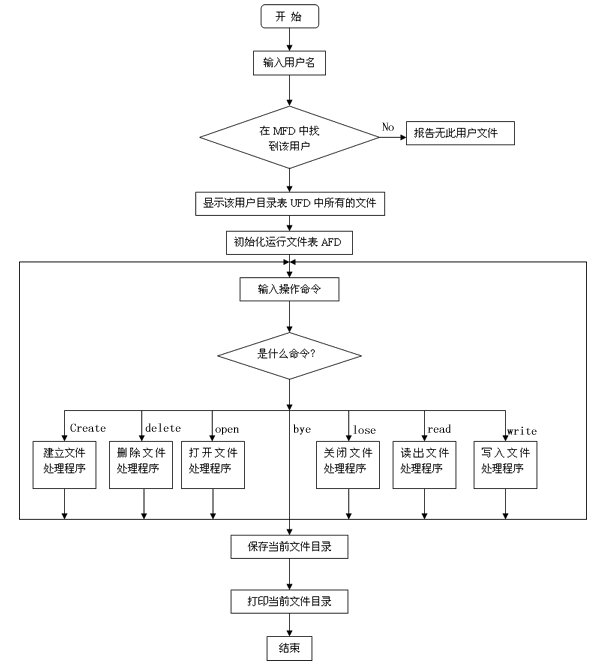
（3）为了便于实现，对文件的读写作了简化，在执行读写命令时，只需改读写指针，并不进行实际的读写操作。

（4）算法与框图

* 因系统小，文件目录的检索使用了简单的线性搜索。
* 文件保护简单使用了三位保护码：允许读写执行、对应位为1，对应位为0，则表示不允许读写、执行。
* 程序中使用的主要设计结构如下：
  + 主文件目录和用户文件目录（MFD、UFD）
  + 打开文件目录（AFD）（即运行文件目录）



文件系统算法的流程图如下：



（5）编一个通过终端选择命令的文件管理系统，每屏要为用户提供足够的选择信息，不需要打入冗长的命令。

**三、实现过程与测试结果分析**

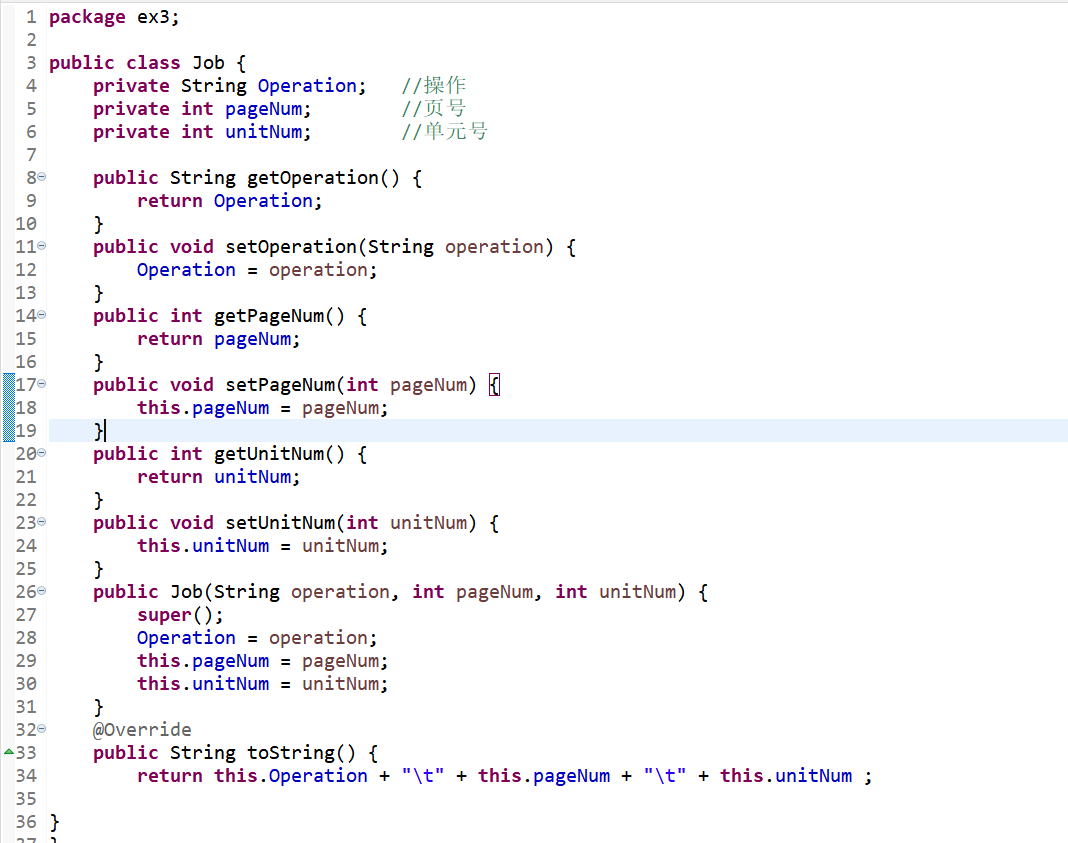
**第一题：模拟分页式存储管理中硬件的地址转换和产生缺页中断**

1. Page类



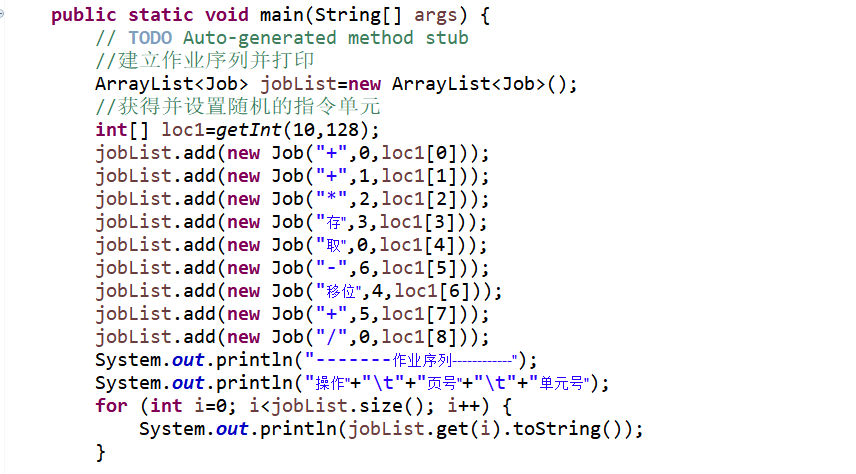
在page类中，定义页号、标记、主存块号、在磁盘位置存储位置。其中当标志位=1，则表示该页已经在主存，标志位=0，则表示该页尚未装入主存。

1. Job类

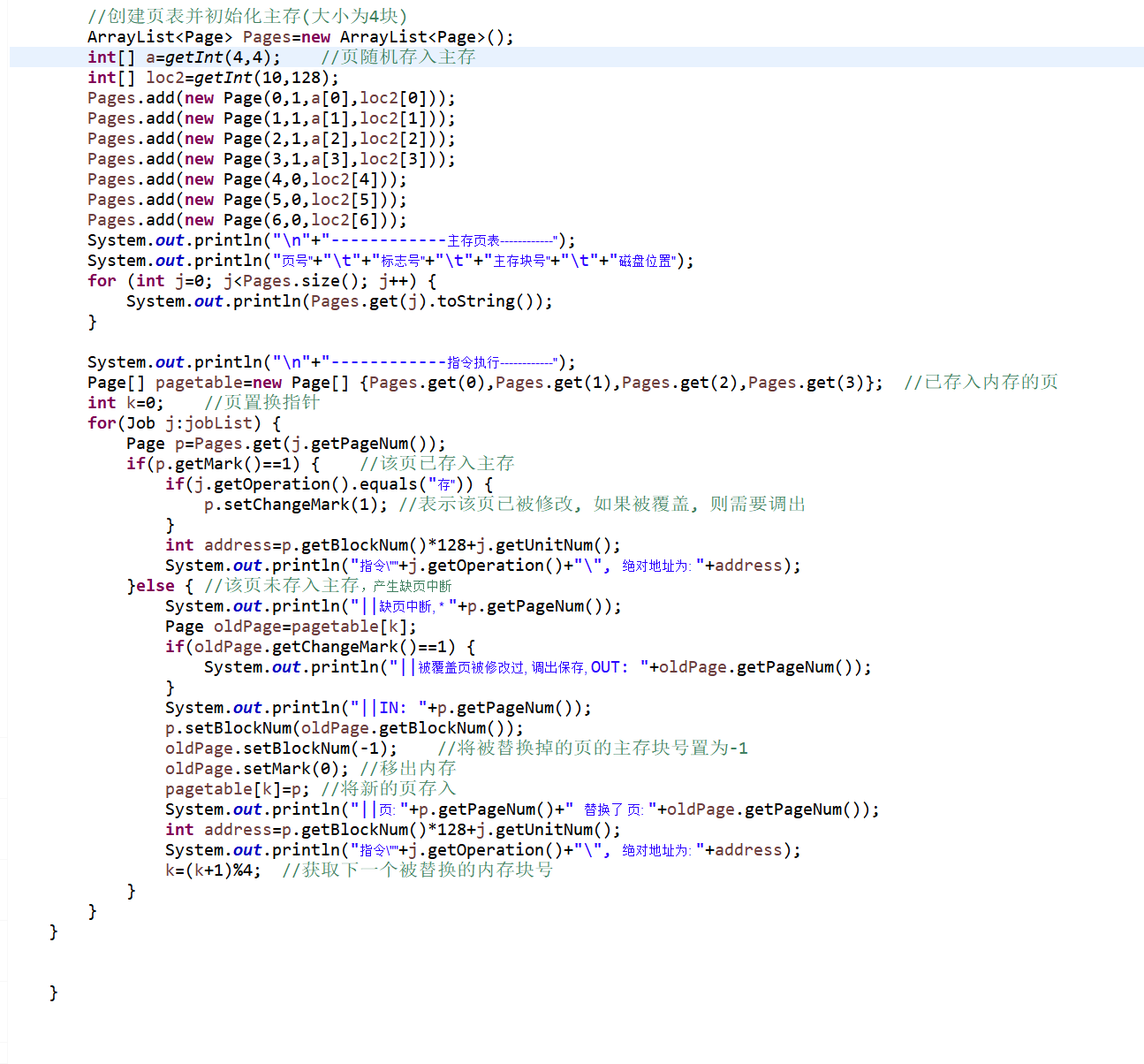


在Job类中，定义操作、页号、单元号

（3）manage类

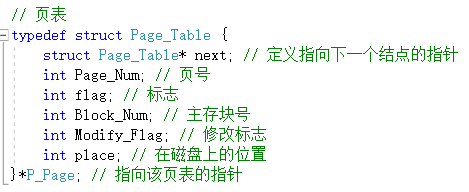


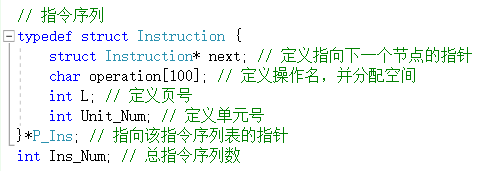
建立作业序列并打印，获得并设置随机的指令单元。



getMark值为1时，表示该页已存为主存，否则表示该页未存入主存，产生了缺页中断；setChangeMark表示该页已被修改，如果被覆盖，则需要调出。

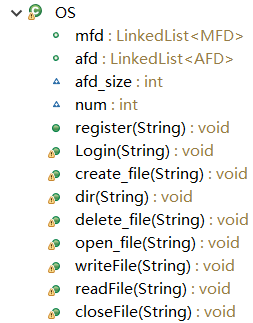
**第二题：用先进先出（FIFO）页面调度算法处理缺页中断**





定义相应页表与指令序列来实现先进先出（FIFO）页面调度算法处理缺页中断。

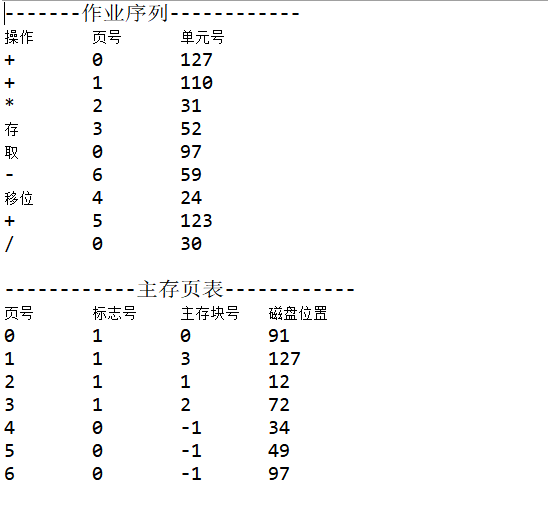
**第四题：用高级语言编写和调试一个简单的文件系统，模拟文件管理的工作过程**



本次实验最主要的类是OS类，实现了对于文件管理的基本的操作。

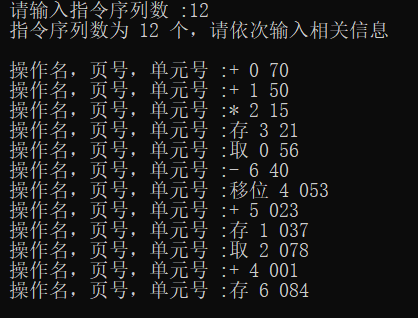
1. **实验结果总结**

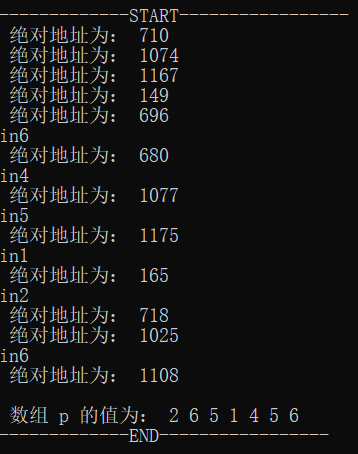
**第一题：模拟分页式存储管理中硬件的地址转换和产生缺页中断**



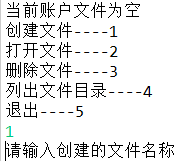


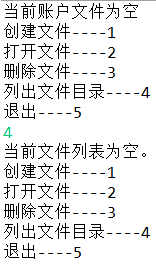
**第二题：用先进先出（FIFO）页面调度算法处理缺页中断**





**第四题：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**





1. **创新的部分**

对文件操作系统实现了创建文件，删除文件，读和写文件，列出文件列表的功能，并限制了用户可以保存和可以运行的文件数量。

**六、对实验的意见与建议**

通过实践操作与理论相结合，更加了解了分页式存储管理中硬件的地址转换和产生缺页中断，加深了对存储管理的作用和工作原理的理解。

通过本次实验，我加深了对FIFO、文件操作系统的理解，并且将老师课堂上讲述的理论知识与实践相结合，收获颇丰。

**七、思考题**

**（1）先进先出页面调度算法的思想？**

FIFO页面置换算法,也就是先进先出的意思。这和我们现实生活中的排队方式很相似，先进队伍的人会先买到票,然后先从队伍中离开。如果使用FIFO算法作为页面置换算法,缓存空间大小是三个页面时,一次进入Page1,Page2,Page3。当Page4要进入缓存时,操作系统将会把Page1清除出缓存,将Page4加入至缓存中。如果再有Page5要进入缓存时,操作系统会将Page2清除出缓存空间,以此类推。

1. **最近最少用（LRU）页面调度算法思想？**

LRU（Least recently used，最近最少使用）算法根据数据的历史访问记录来进行淘汰数据，其核心思想是“如果数据最近被访问过，那么将来被访问的几率也更高”。最常见的实现是使用一个链表保存缓存数据，新数据插入到链表头部；每当缓存命中（即缓存数据被访问），则将数据移到链表头部；当链表满的时候，将链表尾部的数据丢弃。

1. **比较两种调度算法的效率（哪种调度算法使产生缺页中断的次数少）？**

LRU。

1. **分析在什么情况下采用哪种调度算法更有利？**

LRU和FIFO本质都是先进先出的思路，但LRU是针对页面的最近访问时间来进行排序，所以需要在每一次页面访问的时候动态的调整各个页面之间的先后顺序（每一个页面的最近访问时间变了）；而FIFO针对页面进入内存的时间来进行排序，这个时间是固定不变的，所以页面之间的先后顺序是固定不变的。如果程序局部性，则LRU会很好。如果内存中所有页面都没有被访问过会退化为FIFO（如页面进入内存后没有被访问，最近访问时间与进入内存的时间相同）。

**LRU算法性能较好，但系统开销较大；FIFO算法的系统的开销较小，但可能发生Belady现象**

**（5）文件管理和目录管理的思想？**

文件管理是操作系统的五大职能之一，主要涉及文件的逻辑组织和物理组织，目录的结构和管理。所谓文件管理，就是操作系统中实现文件统一管理的一组软件、被管理的文件以及为实施文件管理所需要的一些数据结构的总称(是操作系统中负责存取和管理文件信息的机构)从系统角度来看，文件系统是对文件存储器的存储空间进行组织，分配和回收，负责文件的存储，检索，共享和保护。从用户角度来看，文件系统主要是实现"按名取存"，文件系统的用户只要知道所需文件的文件名，就可存取文件中的信息，而无需知道这些文件究竟存放在什么地方

目录管理：对文件实施有效的管理，必须对它们加以妥善组织，主要是两大操作：基本信息记录(FCB,目录项)，方便检索、管理（目录操作）

目录管理的要求如下：实现“按名存取”；提高对目录的检索速度；文件共享；允许文件重名。文件控制块—FCB：为了能对一个文件进行正确的存取，必须为文件设置用于描述和控制文件的数据结构，称之为“文件控制块”（FCB）。文件与文件控制块一一对应。记录文件名及其存放地址、文件的说明和控制信息。文件管理程序借助于文件控制块中的信息对文件施以各种操作。把文件控制块的有序集合称为文件目录，即一个文件控制块就是一个目录项。通常一个文件目录也被看作是一个文件，称为目录文件。

评分标准：

|  |  |
| --- | --- |
| 考核标准 | 得分 |
| （1）正确理解和掌握实验所涉及的概念和原理（20%）； |  |
| （2）按实验要求合理设计数据结构和程序结构（20%）； |  |
| （3）认真记录实验数据，原理及实验结果分析准确（20%）； |  |
| （4）实验过程中，具有严谨的学习态度和认真、踏实、一丝不苟的科学作风（10%）； |  |
| （5）所做实验具有一定的创新性（10%）； |  |
| （6）实验报告规范（20%）。 |  |