**计科 14 级操作系统课设内容**

**分组完成如下题目,最终提交材料：** 实验程序代码、课程设计报告电子版

**考核要求：** 在计算机上完整运行相应的软件系统程序调试结果，有动态演示 。演示并讲解程序及其运行结果

**题目:**

1. 在windows环境下, 编写程序实现经典进程同步算法
2. 进程调度模拟程序设计（进程管理）
3. 模拟实现动态分区存储管理（存储管理）
4. 分页管理方式下存储分配情况模拟（存储管理）
5. 页面置换模拟程序设计（存储管理）
6. Socket通信实验 （进程间通信）

**题目具体要求:**

1. **windows环境下编写程序实现经典进程同步算法**

实现下列问题的同步,并选择其中一个给出包含界面的动态展示,参考示例(如下图)：放电影问题的运行界面之一

****

1)哲学家进餐问题(问题描述见教材)

2)理发师睡觉问题

理发店里有一位理发师，一把理发椅和N把供等候理发的顾客坐的椅子

如果没有顾客,则理发师便在理发椅上睡觉。当一个顾客到来时，他必须先唤醒理发师。如果顾客到来时理发师正在理发，则如果有空椅子，可坐下来等；否则离开。

3)医生看病问题

三个病椅，十个等椅，三个医生。

病人来了以后，若病椅有空闲，直接去看病；

若病椅满，而等椅有空闲，坐在等椅上等待；

若病椅和等椅均满，则离开。

若无病人看病，医生等待。

给出医生和病人的程序，正确实现互斥和同步。

4)吸烟问题

一个系统，有三个吸烟者(smoker)进程和一个供应商(agent)进程。每个吸烟者制造并吸掉香烟。但制造和吸掉香烟需要三种原料：烟草、纸和火柴。供货商有充足的原料，而每个吸烟者手中只有一种不同于其它吸烟者所拥有的原料。供货商放两种原料于桌面上，于是正好有另一种原料的吸烟者就可以制造并吸掉香烟，然后发一个完成信号给供货商，于是供货商马上给出另两种原料，并重复这个过程。请写出供货商和吸烟者的同步算法。

5)放电影问题

一个电影院中有3个放映厅，每个放映厅中可以放映电影院中有的所以影片。如果顾客来到电影院中，某放映厅中正在放映他所需要看的影片，他就可以马上进入放映厅。如果没有一个放映厅放映他所想看的影片，他就需要排队等候。放映厅中只要有个顾客还在看，影片将循环放映下去，直到所有的顾客都离开了，再由正在排队的下一位顾客选择需要放映的下一部影片。请用信号量编程实现此电影院的管理。

# 2.进程调度模拟程序设计（进程管理）

1．给出进程调度的算法描述（如基于动态优先级和时间片轮转调度算法的描述）。

2．用C语言设计一个对n个并发进程进行调度的程序，每个进程由一个进程控制块（PCB）结构表示，该进程控制块应包括下述信息：进程标识ID、进程优先数PRIORITY（并规定优先数与优先权成正比）、时间片数CHIP、进程已经占用CPU的时间CPUTIME，进程还需要运行的时间ALLTIME（当进程运行完毕时，其值为0）、进程的状态STATE（为简化起见。设每个进程处于运行E（excecuting）、就绪R（ready）和完成F（finish）三种状态之一，并假设起始状态都是就绪状态R。），以及进程队列指针NEXT（用来将PCB排成队列）等，可按照调度算法的不同而增删。

3．调度程序应当包含2种不同的调度算法，运行时可以任选一种，以利于各种方法的分析和比较。

4．程序应能显示或打印各种进程状态和参数变化情况，便于观察。即要显示每个时间片内各进程的情况，并且指出运行进程及就绪和阻塞队列中的内容。

**(1)按优先数调度算法实现处理器调度的程序。**

[提示]：

(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，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。

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”。

要求运行时间——假设进程需要运行的单位时间数。

优先数——赋予进程的优先数，调度时总是选取优先数大的进程先执行。

状态——可假设有两种状态，“就绪”状态和“结束”状态。五个进程的初始状态都为“就绪”，用“R”表示，当一个进程运行结束后，它的状态为“结束”，用“E”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。例：

队首标志

K2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K1 | P1 | K2 | P2 | K3 | P3 | K4 | P4 | K5 | P5 |
|  | 0 |  | K4 |  | K5 |  | K3 |  | K1 |
|  | 2 |  | 3 |  | 1 |  | 2 |  | 4 |
|  | 1 |  | 5 |  | 3 |  | 4 |  | 2 |
|  | R |  | R |  | R |  | R |  | R |
|  | PCB1 |  | PCB2 |  | PCB3 |  | PCB4 |  | PCB5 |

(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。

**(2)按时间片轮转法实现处理器调度的程序。**

[提示]：

(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，进程名——作为进程的标识，假设五个进程的进程名分别为Q1，Q2，Q3，Q4，Q5。

指针——进程按顺序排成循环队列，用指针指出下一个进程的进程控制块的首地址，最后一个进程的指针指出第一个进程的进程控制块首地址。

要求运行时间——假设进程需要运行的单位时间数。

已运行时间——假设进程已经运行的单位时间数，初始值为“0”。

状态——有两种状态，“就绪”和“结束”，初始状态都为“就绪”，用“R”表示。当一个进程运行结束后，它的状态为“结束”，用“E”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。例如，当前轮到P2执行，则有：

标志单元

K2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K1 | Q1 | K2 | Q2 | K3 | Q3 | K4 | Q4 | K5 | Q5 |
|  | K2 |  | K3 |  | K4 |  | K5 |  | K1 |
|  | 2 |  | 3 |  | 1 |  | 2 |  | 4 |
|  | 1 |  | 0 |  | 0 |  | 0 |  | 0 |
|  | R |  | R |  | R |  | R |  | R |
|  | PCB1 |  | PCB2 |  | PCB3 |  | PCB4 |  | PCB5 |

(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。

请同学注意：在实际的系统中，当一个进程被选中运行时，必须置上该进程可以运行的时间片值，以及恢复进程的现场，让它占有处理器运行，直到出现等待事件或运行满一个时间片。在这时省去了这些工作，仅用“已运行时间+1”来表示进程已经运行满一个时间片。

(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≠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。

# 3.模拟实现动态分区存储管理

编程实现动态分区存储管理方式的主存分配与回收。具体内容包括：首先确定主存空间分配表；然后采用最优适应算法及首次适应算法完成主存空间的分配和回收。

具体讲：

初始状态：动态分区管理方式预先不将主存划分区域。而是把主存除操作系统占用区域外的空间看作一个大的空闲区。当作业要求装入主存时，根据作业的大小查询主存内各空闲区。并按照特定的算法选择一合适的空闲区，按作业大小的要求画出一个分区并装入该作业。剩下的区域作为新的空闲区。

当作业执行完毕后，所占用的主存空间将被回收，成为一个空闲区。注意如果该空闲区的相邻分区也是空闲区，则需要将相邻空闲区合并成一个空闲区。

提示：

动态分区大小由作业需求量决定，分区的长度预先不能固定。

可建立两张分区表记录主存使用情况：

“已分配表”记录作业占用分区；“空闲区表”记录空闲区。

主程序菜单可参考：

选择功能项（0-退出、1-分配主存、2-回收主存、3-显示主存）

分配时，要求输入作业名和长度

回收时，要求输入要回收的作业名

显示主存，则显示空闲分区情况以及已分配分区情况等。

本实验模拟在两种存储管理方式下的主存分配和回收。

图例：

可变分区方式是按作业需要的主存空间大小来分割分区的。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。随着作业的装入、撤离，主存空间被分成许多个分区，有的分区被作业占用，而有的分区是空闲的。例如：

|  |
| --- |
| 操作系统  0  5k  10k  14k  26k  32k  128k |
| 作业1 |
| 作业3 |
| 空闲区 |
| 作业2 |
| 空闲区 |

为了说明哪些区是空闲的，可以用来装入新作业，必须要有一张空闲区说明表，格式如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 起 址 | 长 度 | 状 态 |
| 第一栏 | 14 K | 12 K | 未 分 配 |
| 第二栏 | 32 K | 96 K | 未 分 配 |
|  |  |  | 空 表 目 |
|  |  | 空 表 目 |
|  |  |  |

其中，起址——指出一个空闲区的主存起始地址。

长度——指出从起始地址开始的一个连续空闲的长度。

状态——有两种状态，一种是“未分配”状态，指出对应的由起址指出的某个长度的区域是空闲区；另一种是“空表目”状态，表示表中对应的登记项目是空白（无效），可用来登记新的空闲区（例如，作业撤离后，它所占的区域就成了空闲区，应找一个“空表目”栏登记归还区的起址和长度且修改状态）。由于分区的个数不定，所以空闲区说明表中应有适量的状态为“空表目”的登记栏目，否则造成表格“溢出”无法登记。

上述的这张说明表的登记情况是按提示（1）中的例所装入的三个作业占用的主存区域后填写的。

(2) 当有一个新作业要求装入主存时，必须查空闲区说明表，从中找出一个足够大的空闲区。有时找到的空闲区可能大于作业需要量，这时应把原来的空闲区变成两部分：一部分分给作业占用；另一部分又成为一个较小的空闲区。为了尽量减少由于分割造成的空闲区，而尽量保存高地址部分有较大的连续空闲区域，以利于大型作业的装入。为此，在空闲区说明表中，把每个空闲区按其地址顺序登记，即每个后继的空闲区其起始地址总是比前者大。为了方便查找还可使表格“紧缩”，总是让“空表目”栏集中在表格的后部。

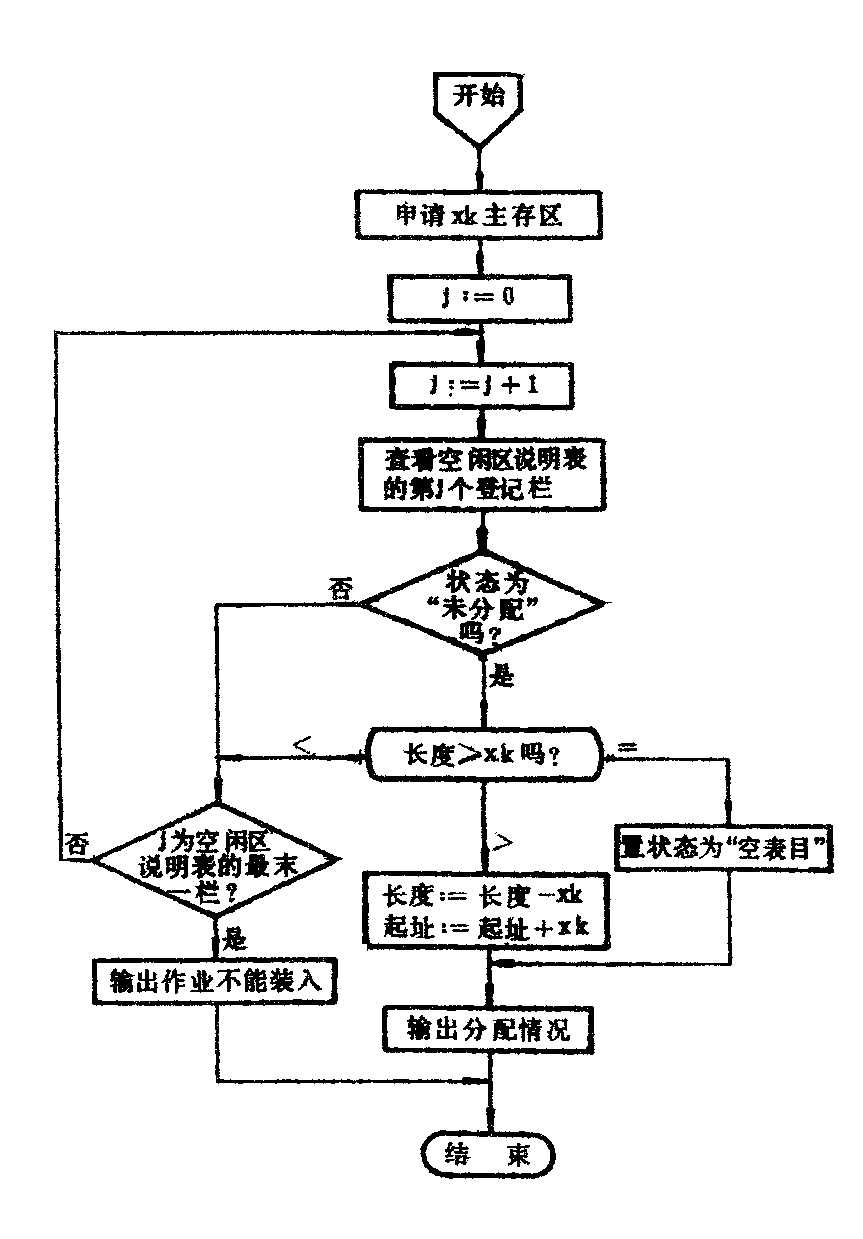
(3) 采用最先适应算法（顺序分配算法）分配主存空间。

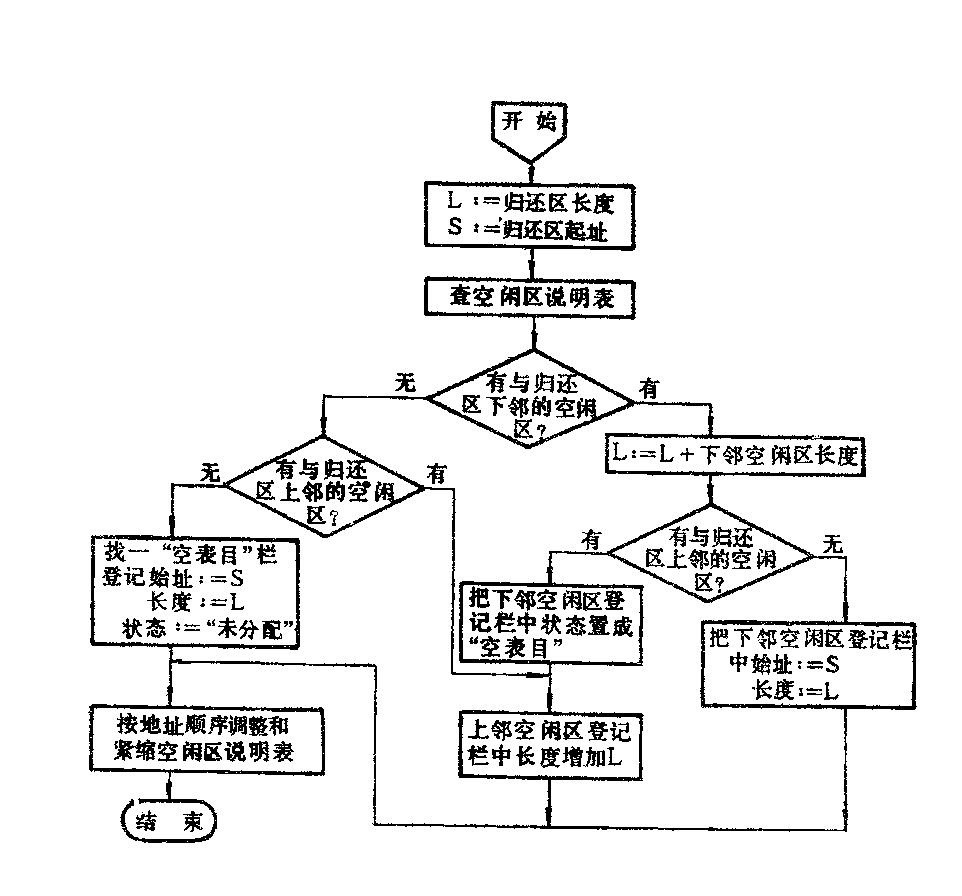
按照作业的需要量，查空闲区说明表，顺序查看登记栏，找到第一个能满足要求的空闲区。当空闲区大于需要量时，一部分用来装入作业，另一部分仍为空闲区登记在空闲区说明表中。

由于本实验是模拟主存的分配，所以把主存区分配给作业后并不实际启动装入程序装入作业，而用输出“分配情况”来代替。最先适应分配算法如图4-1。

(4) 当一个作业执行结束撤离时，作业所占的区域应该归还，归还的区域如果与其它空闲区相邻，则应合成一个较大的空闲区，登记在空闲区说明表中。例如，在提示（1）中列举的情况下，如果作业2撤离，归还所占主存区域时，应与上、下相邻的空闲区一起合成一个大的空闲区登记在空闲区说明表中。归还主存时的回收算法如图4-2。

(5) 请按最先适应算法设计主存分配和回收的程序。然后按（1）中假设主存中已装入三个作业，且形成两个空闲区，确定空闲区说明表的初值。现有一个需要主存量为6K的作业4申请装入主存；然后作业3撤离；再作业2撤离。请你为它们进行主存分配和回收，把空闲区说明表的初值以及每次分配或回收后的变化显示出来或打印出来。

图4-1 最先适应分配模拟算法

图4-2 主存回收算法

# 4.分页管理方式下存储分配情况模拟

借用外存空闲分区管理中位示图的方法来表示主存分配情况，实现主存空间的分配和回收。

[提示]：

(1) 分页式存储器把主存分成大小相等的若干块，作业的信息也按块的大小分页，作业装入主存时可把作业的信息按页分散存放在主存的空闲块中，为了说明主存中哪些块已经被占用，哪些块是尚未分配的空闲块，可用一张位示图来指出。位示图可由若干存储单元来构成，其中每一位与一个物理块对应，用0/1表示对应块为空闲/已占用。

(2) 假设某系统的主存被分成大小相等的64块，则位示图可用8个字节来构成，另用一单元记录当前空闲块数。如果已有第0，1，4，5，6，9，11，13，24，31，共10个主存块被占用了，那么位示图情况如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字 位  节 数  号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

(3) 当要装入一个作业时，根据作业对主存的需要量，先查当前空闲块数是否能满足作业要求，若不能满足则输出分配不成功。若能满足，则查位示图，找出为“0”的一些位，置上占用标志“1”，从“当前空闲块数”中减去本次占用块数。

按找到的计算出对应的块号，其计算公式为：

块号= j×8+i

其中，j表示找到的是第n个字节，I表示对应的是第n位。

根据分配给作业的块号，为作业建立一张页表，页表格式：

|  |  |
| --- | --- |
| 页 号 | 块 号 |
| 0 |  |
| 1 |  |
| 2 |  |
|  |  |

(4) 当一个作业执行结束，归还主存时，根据该作业的页表可以知道应归还的块号，由块号可计算出在位示图中的对应位置，把对应位的占用标志清成“0”，表示对应的块已成为空闲块。归还的块数加入到当前空闲块数中。由块号计算在位示图中的位置的公式如下：

字节号 j=[块号/8] （[ ]表示取整）

位数 i={块号/8} （{ }表示取余）

(5) 设计实现主存分配和回收的程序。假定位示图的初始状态如（2）所述，现有一信息量为5页的作业要装入，运行你所设计的分配程序，为作业分配主存且建立页表（格式如（3）所述）。然后假定有另一作业执行结束，它占用的块号为第4，5，6和31块，运行你所设计的回收程序，收回作业归还的主存块。

要求能显示和打印分配或回收前后的位示图和当前空闲块数，对完成一次分配后还要显示或打印为作业建立的页表。

# 5.页面置换模拟程序设计（存储管理）

    1．假设分给一作业的内存块数为4，每个页面中可存放10条指令。

    2．用C语言设计一个程序，模拟一作业的执行过程。设该作业共有320条指令，即它的地址空间为32页，目前它的所有页面都还未调入内存。在模拟过程中，如果所访问的指令已经在内存，则显示其物理地址，并转下一条指令。如果所访问的指令尚未装入内存，则发生缺页，此时需记录缺页的次数，并将相应页调入内存。如果4个内存块中均已装入该作业的虚页面，则需进行页面置换。最后显示其物理地址，并转下一条指令。在所有320条指令执行完毕后，请计算并显示作业运行过程中发生的缺页率。

  3．置换算法：请分别考虑OPT、FIFO、LRU、LFU和NRU算法。

   4．作业中指令的访问次序要求按下述原则生成：

 50%的指令是顺序执行的。

 25%的指令是均匀分布在前地址（即低地址）部分。

 25%的指令是均匀分布在后地址（即高地址）部分。

具体的实施办法是：

    ① 在[0，319]之间随机选取一条起始执行指令，其序号为m；

② 顺序执行下一条指令，即序号为m+1的指令；

③ 通过随机数，跳转到前地址部分[0，m-1]中的某条指令处，其序号为m1；

④ 顺序执行下一条指令，即序号为m1+1的指令；

⑤ 通过随机数，跳转到后地址部分[m1+2，319]中的某条指令处，其序号为m2；

⑥ 顺序执行下一条指令，即序号为m2+1的指令；

⑦ 重复“跳转到前地址部分、顺序执行、跳转到后地址部分、顺序执行”的过程，直至执行完全部320条指令。

5.指令序列转换成页访问序列。

第 0 条-第 9 条指令为第0页（对应虚存地址为[0，9]）

第10条-第19条指令为第1页（对应虚存地址为[10，19]）

………………………………

第310条-第319条指令为第31页（对应虚存地址为[310，319]）

# 6.Socket通信实验（进程间通信）

编写程序，在windows下使用C语言设计和实现一个简单的基于Socket通信的程序。 一、实验目的：

1.理解Socket通信的基本概念和工作原理；

2.掌握Socket的建立、监听、连接、发送和接收数据的方法；

二、实验内容：

1.在同一台机器上建立一个服务器端Server，一个客户端Client；

2.建立服务器端Server与客户端Client的连接；

3.服务器端Server提供一个计算功能，进行四则运算(除号用/代替)；Client从键盘接收运算格式串，串的格式为：操作符，操作数1，操作数2 例如发送字符串：\*，3，5表示计算3\*5

4. Server接收命令串，解析并计算之，结果以字符串形式返回给Client；

5. Client显示结果；

6.输入q结束Client，Client结束前断开与Server的连接。

三、准备知识 1.了解Socket通信的基本概念和工作原理； 2.了解必要的网络知识，如TCP协议等.