页面置换算法的实验报告

操作系统

课程设计报告

院　（系）：衡阳师范学院

专　　业：计算机科学与技术

姓　　名：陈建元　　齐欢

班　　级：1103 班

学　　号：11190301　11190316

题　　目：页面置换算法

指导教师：王玉奇

2013 年 12 月 10 日至 12 月 28 日

目　　录

摘　要

操作系统（英语；Operating System，简称OS）是一管理电脑硬件与软件资源的程序，同时也是计算机系统的内核与基石。操作系统身负诸如管理与配置内存、决定系统资源供需的优先次序、控制输入与输出设备、操作网络与管理文件系统等基本事务。操作系统是管理计算机系统的全部硬件资源包括软件资源及数据资源;控制程序运行;改善人机界面;为其它应用软件提供支持等，使计算机系统所有资源最大限度地发挥作用，为用户提供方便的、有效的、友善的服务界面。操作系统是一个庞大的管理控制程序，大致包括 5 个方面的管理功能:进程与处理机管理、作业管理、存储管理、设备管理、文件管理。　在地址映射过程中，若在页面中发现所要访问的页面不再内存中，则产生缺页中断。当发生缺页中断时操作系统必须在内存选择一个页面将其移出内 存，以便为即将调入的页面让出空间。而用来选择淘汰哪一页的规则叫做页面置换算法（Page-Replacement Algorithms）。

关键词：操作系统；OPT 页面置换算法；FIFO 先进先出的算法；LRU 最近最久未使用夜面置换算法

第一章　设计任务和需求

1.1 课程设计任务

深入掌握内存调度算法的概念原理和实现方法。

编写程序实现：

（1）　先进先出页面置换算法（FIFO）

（2）　最近最久未使用页面置换算法（LRU）

（3）　最佳置换页面置换算法（OPT）

设计一个虚拟存储区和内存工作区，编程序演示以上三种算法的具体实现过程，并计算访问命中率。演示页面置换的三种算法。通过随机数产生一个指令序列，将指令序列转换成为页地址流。计算并输出各种算法在不同内存容量下的缺页率。

1.2 课程设计需求

在各种存储器管理方式中，有一个共同的特点，即它们都要求将一个作业全部装入内存方能运行，但是有两种情况：（1） 有的作业很大，不能全部装入内存，致使作业无法运行；（2） 有大量作业要求运行，但内存容量不足以容纳

所有这些作业。而虚拟内存技术正式从逻辑上扩充内存容量，将会解决以上两个问题。　从内存中调出一页程序或数据送磁盘的对换区中，通常，把选择换出的页面的算法称为页面置换算法（Page-Replacement Algorithms ）。进而页面置换算法程序能客观的将其工作原理展现在我们面前。

第二章　概要设计

2.1 系统分析

由于分区式管理尽管实现方式较为简单，但存在着严重的碎片问题使得内存的利用率不高。再者，分区式管理时，由于各作业或进程对应于不同的分区以及在分区内各作业或进程连续存放，进程的大小或内存可用空间的限制。而且分区式管理也不利于程序段和数据段的共享。页式管理正是为了减少碎片以及为了只在内存存放那些反复执行或即将执行的程序段与数据部分，而把那些不经常执行的程序段和数据存放于外存待执行时调入，以提高内存利用率而提出来的页式管理有动态页式管理和静态页式管理之分，动态页式管理是在静态页式管理的基础上发展起来的。请求页式管理属于动态页式管理中的一种，它的地址变换过程与静态页式管理时的相同，也是通过页表查出相应的页面号，由页面号与页内相对地址相加而得到实际物理地址。有关的地址变换部分是由硬件自动完成的。当硬件变换机构发现所要求的页不在内存时，产生缺页中断信号，由中断处理程序做出相应的处理。中断处理程序是由软件实现的。除了在没有空闲页面时要按照置换算法选择出被淘汰页面之外，还要从外存读入所需要的虚页。这个过程要启动相应的外存和涉及到文件系统。因此，请求页式管理是一个十分复杂的过程，内存利用率的提高是以牺牲系统开销的代价换来的。这里用到了置换算法。它是在内存中没有空闲页面时被调用。目的是选出一个被淘汰的页面。如果内存中有足够的空闲页面存放所调入的页，则不必使用置换算法。把内存和外存统一管理的真正目的是把那些被访问概率非常高的页存放在内存中。因此，置换算法应该置换那些被访问概率低的页，将它们移出内存。

2.2 调页策略

2.2.1何时调入页面

如果进程的许多页是存放在外存的一个连续区域中，则一次调入若干个相邻的页，会比一次调入一页的效率更高效一些。但如果调入的一批页面中的大多数都未被访问，则又是低效的。可采用一种以预测为基础的预调页策略，将那些预计在不久之后便会被访问的页面，预先调入内存。如果预测较准确，那么，这种策略显然是很有吸引力的。但目前预调页的成功率仅为 50%。且这种策略主要用于进程的首次调入时，由程序员指出应该先调入哪些页。

2.2.2请求调页策略

当进程在运行中需要访问某部分程序和数据时，若发现其所在的页面不在内存，便即提出请求，由 OS将其所需页面调入内存。由请示调页策略所确定调入的页，是一定会被访问的，再加之请求调页策略比较易于实现，故在目前的虚拟存储器中，大多采用此策略。但这种策略每次仅调入一页，故须花费较大的系统开销，增加了磁盘 I/O 的启用频率。

2.2.3从何处调入页面

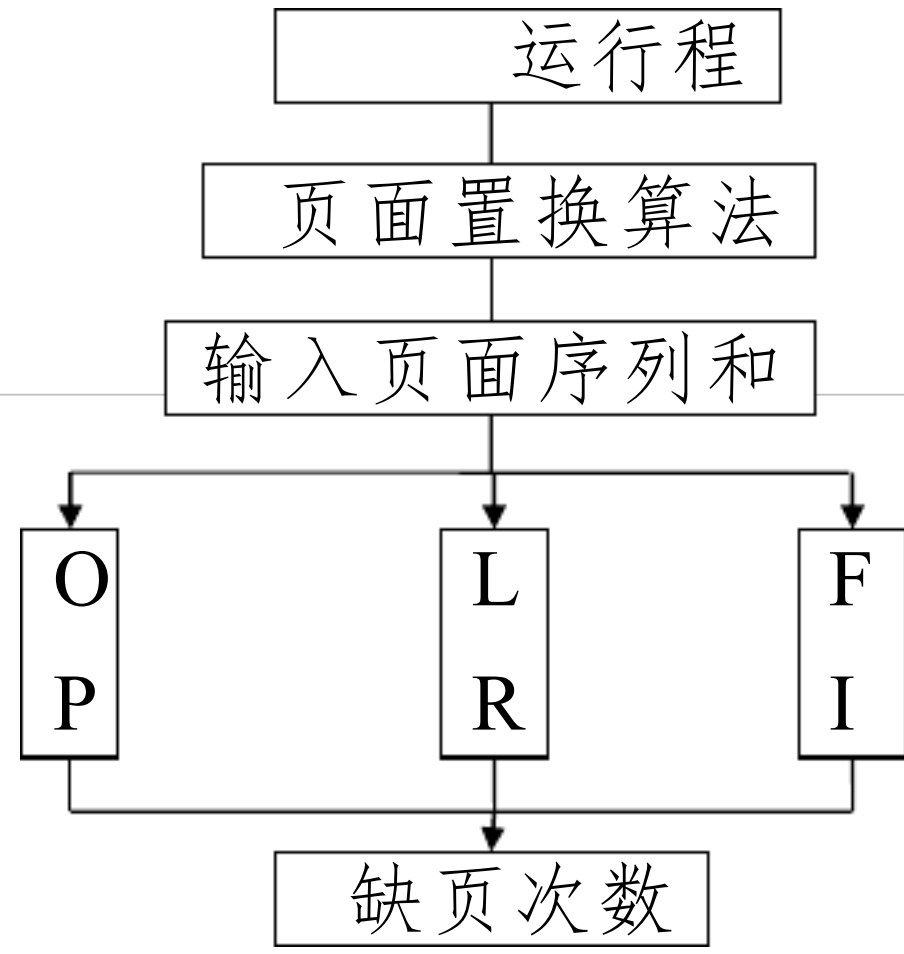
在请求分页系统中的外存分为两部分：用于存放文件的文件区和用于存放对换页面的对换区。通常，由于对换区是采用连续分配方式，而事件是采用离散分配方式，故对换区的磁盘 I/O 速度比文件区的高。这样，每当发生缺页请求时，系统应从何处将缺页调入内存，可分成如下三种情况：

① 系统拥有足够的对换区空间，这时可以全部从对换区调入 所需页面，以提高调页速度。为此，在进程运行前，便须将与该进程有关的文件，从文件区拷贝到对换区。

② 系统缺少足够的对换区空间，这时凡是不会被修改的文件，都直接从文件区调入；而当换出这些页面时，由于它们未被修改而不必再将它们换出时，以后需要时，再从对换区调入。

③ UNIX方式。由于与进程有关的文件都放在文件区，故凡是未运行过的页面，都应从文件区调入。而对于曾经运行过但又被换出的页面，由于是被放在对换区，因此在下次调入时，应从对换区调入。由于 UNIX系统允许页面共享，因此，某进程所请求的页面有可能已被其它进程调入内存，此时也就无须再从对换区调入。

2.3 模块设计



详细设计

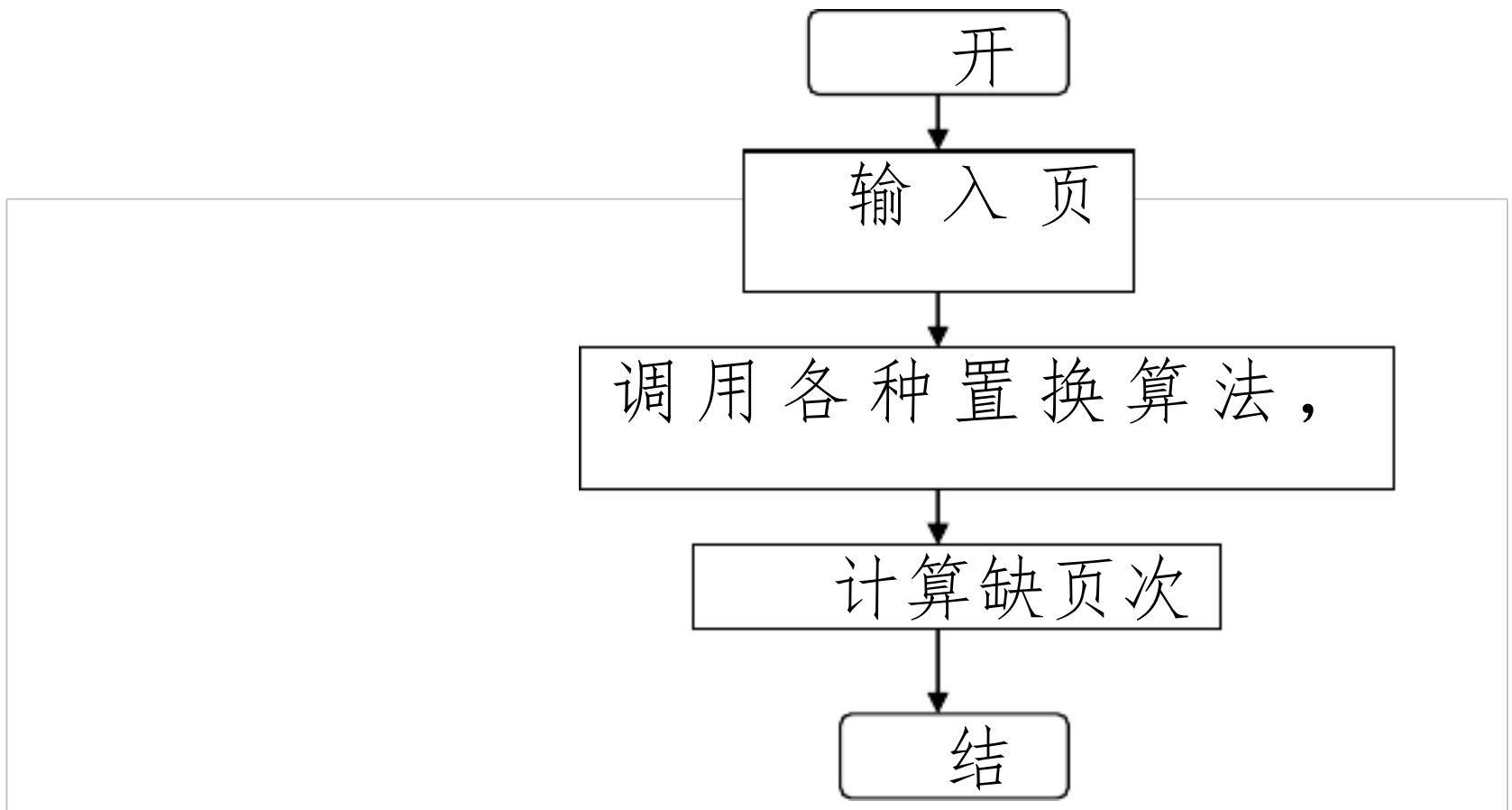
系统设计

在进程运行过程中，若其所要访问的页面不在内存而需把它们调入内存，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数据，送磁盘的对换区中。但应将哪个页面调出，须根据一定的算法来确定。通常，把选择换出页面的算法称为页面置换算法(Page_Replacement Algorithms)。一个好的页面置换算法，应具有较低的页面更换频率。从理论上讲，应将那些以后不再会访问的页面换出，或将那些在较长时间内不会再访问的页面调出。

3.2 算法思想及流程图

3.2.1 主程序流程图

如图（3—2—1）所示

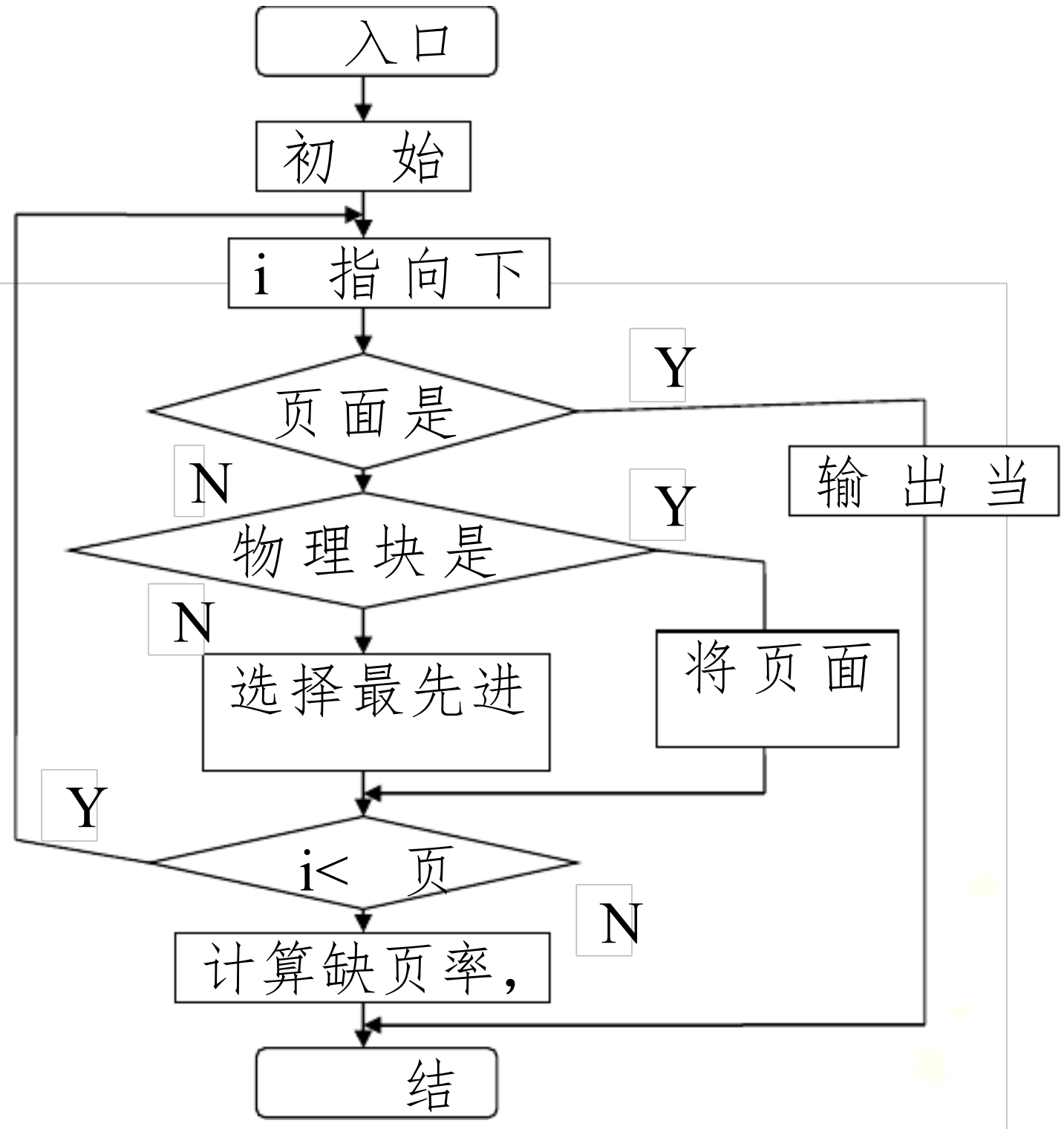


主流程图（图 —2—1）

(FIFO) 页面置换算法

算法的基本思想：这是最早出现的置换算法。该算法总是淘汰最先进入内存的页面，即选择在内存中驻留时间最久的页面予以淘汰。该算法实现简单只需把一个进程已调入内存的页面，按先后次序存入一个时间数组，并将其中时间值最大的页面进行淘汰，并替换入新的页面就可以实现。

算法流程图：如图（3—2—2）所示

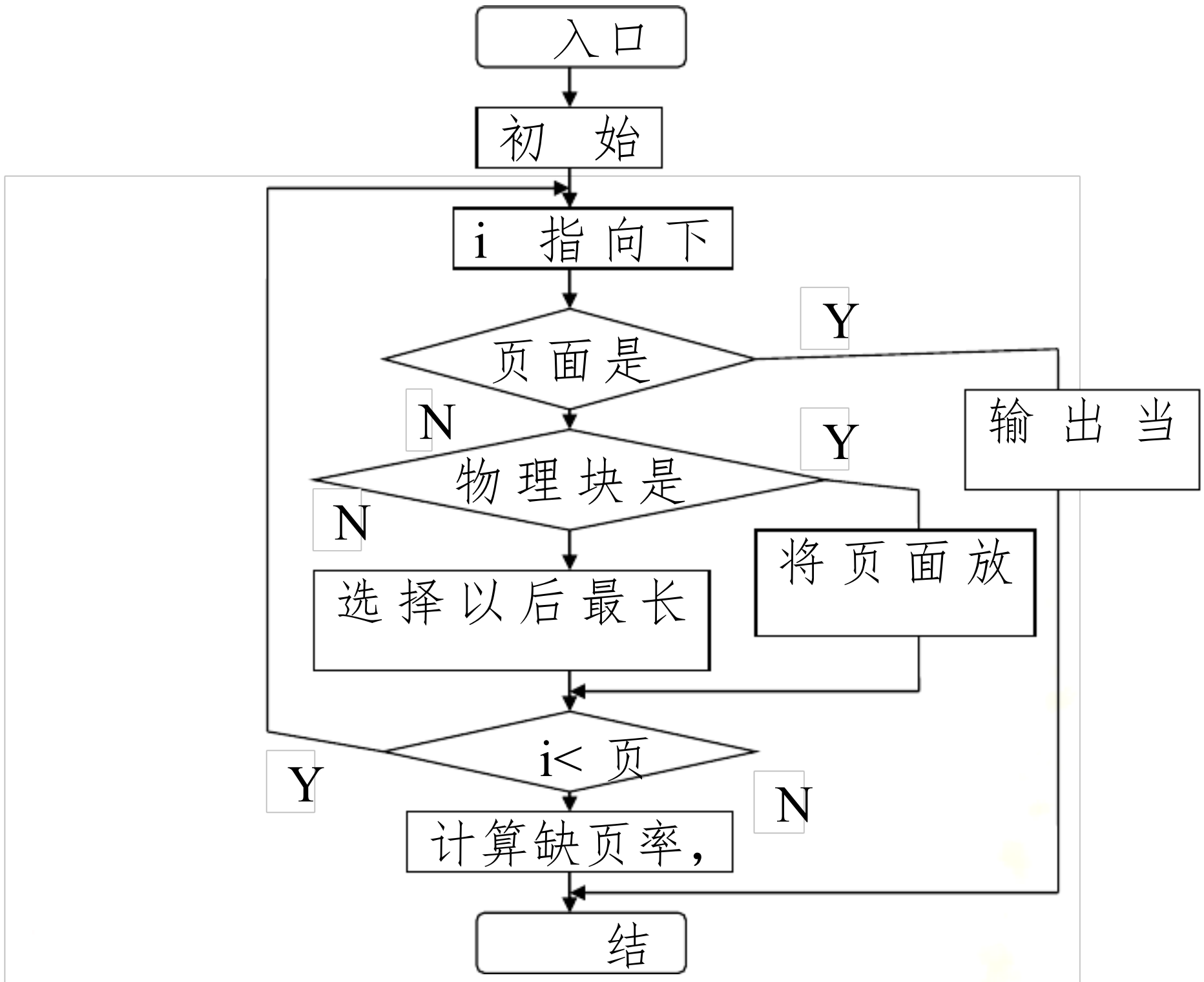


FIFO置换算法（图 —2—2）

OPT）

算法的基本思想：其所选择的被淘汰页面，将是永不使用的，或者是在最长时间内不再被访问的页面。可保证获得最低的缺页率。但由于人们目前还无法预知一个进程在内存的若干个页面中，哪一个页面是未来最长时间内不再被访问的，因而该算法也是无法实现的。但是可利用该算法去评价其它算法。

算法流程图：如图（3—2—3）所示

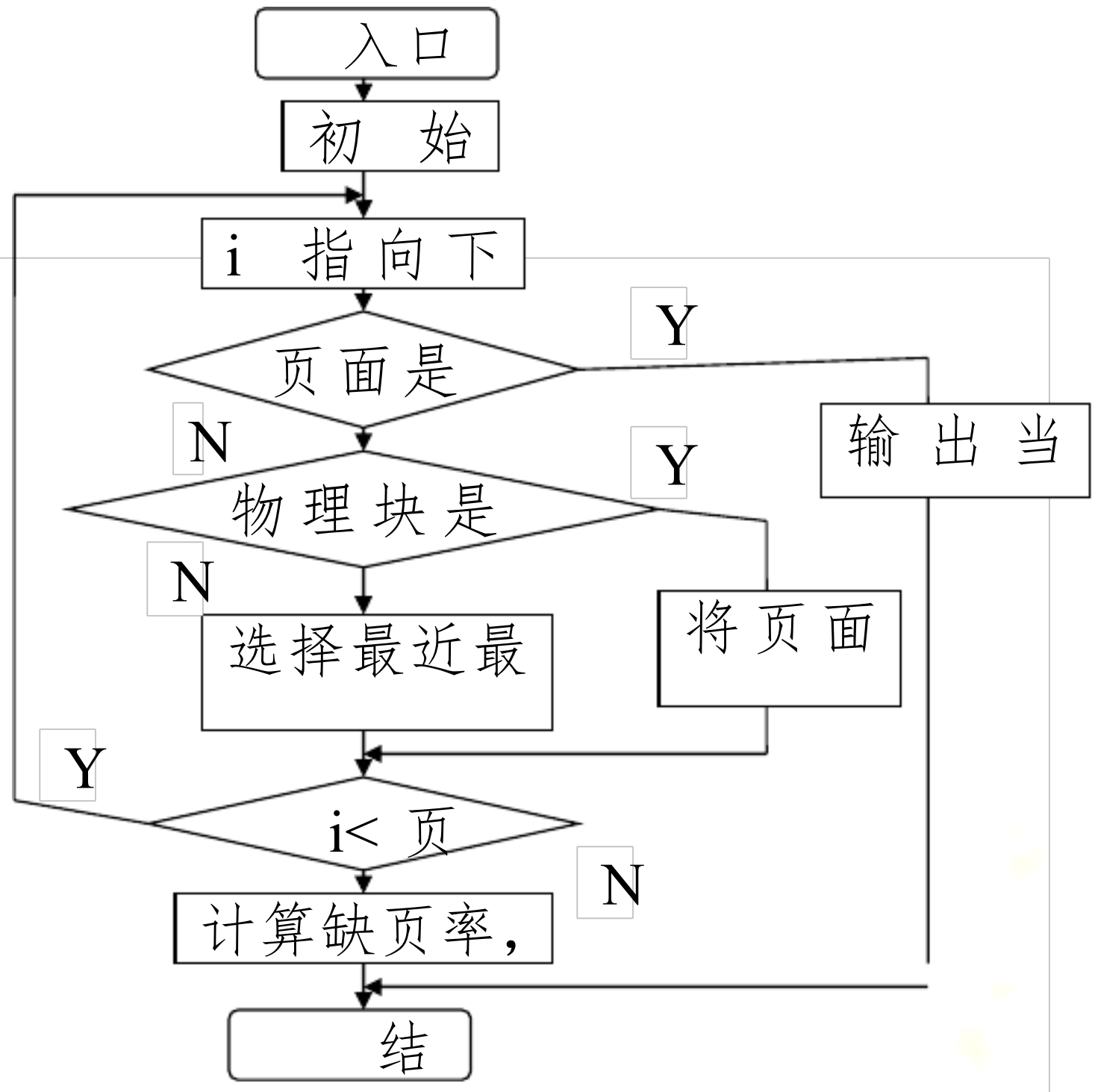


OPT页面置换算法（图 —2—3）

LRU

算法的基本思想：当需要淘汰某一页时，选择离当前时间最近的一段时间内最久没有使用过的页先淘汰。该算法的主要出发点是，如果某页被访问了，则它可能马上还被访问。或者反过来说，如果某页很长时间未被访问，则它在最近一段时间不会被访问。

算法流程图：如图（3—2—4）所示



LRU页面置换算法（图 —2—4）

源程序结构分析

程序结构

Input(int m,Pro p[L])// 打印页面走向状态

srand(j);// 以时钟时间j为种子,初始化随机数发生器

void print(Pro *page1)// 打印当前的页面

int Search(int e,Pro *page1)/寻找内存块中与 e 相同的块号

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/838012125126006076