

数据结构与算法分析（2017 年秋季）

实验一 数组与链表

一、 实验时间和地点

时间: 2016 年 10 月 10 日(第 6 周周二)下午 2 点到 4 点半; 地点: **B7-231, 233**

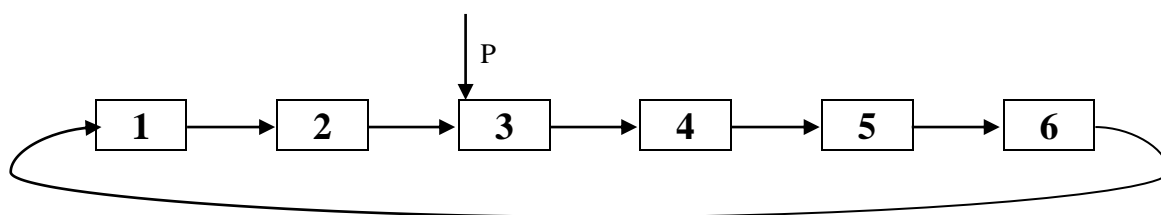
二、 实验目的

掌握数组以及链表结构的算法实现与应用。

三、 实验内容

利用循环数组或者单向循环链表结构实现如下内存页置换策略 CLOCK Algorithm:

假设缓冲池有 6 个缓冲页, 以单向循环链表结构组织, 结构如下图所示。缓冲池初始状态: 所有缓冲页可用, 指针 P 指向下一个即将被分配的缓冲页。缓冲池包含的缓冲页数作为程序运行参数设置, 该参数的有效设置范围为[5, 10]。



生成一组随机物理页访问序列, 页编号范围是 0 到 19, 页面访问序列长度作为程序运行参数设置。如: 9, 1, 10, 1, 9, 18, 7, 7, 12, 17 为一个长度为 10 的物理页访问序列。

对每一次物理页访问, 若该页已经在缓冲池中, 则显示 Hit 并将该页对应的缓冲页状态更改为 $R=1$; 若该页不在缓冲池中, 则分配一个缓冲页并将对应缓冲页的状态设为 $R=0$;

当需要分配一个缓冲页给当前访问的物理页时: 若指针 P 指向的缓冲页状态为 $R=1$, 表示该页刚刚被访问过应该继续保留在缓冲池中, 因此将该页的状态更改为 $R=0$, 并使 P 指向下一个缓冲页; 若指针 P 指向的缓冲页状态为 $R=0$, 表明该页已经有较长时间没有被访问, 可以将该缓冲页分配给当前访问的物理页。

四、 实验基础知识

1. C++编程技术

五、 主要实验步骤

1. 分析 CLOCK 算法设计需求，定义合适的数组或者链表结构；
2. 实现物理页的随机访问序列函数 `genSeq()`；
3. 分析页置换策略，分别实现物理页访问函数 `visitPage()`以及缓冲页分配函数 `allocBuff()`；
4. 整合关键函数，设计并实现 CLOCK 页置换策略函数 `ClockPR(int buf_len, int seq_len)`。

六、 实验报告

1. 提交纸质版和电子版实验报告、以及源程序代码。（实验报告按模板撰写）
 2. 纸质版实验报告提交截止时间为 2016 年 10 月 17 日（周二）上午 10 点 40 分。
 3. 将电子版实验报告和代码打包并命名为“学号_姓名_lab1.zip”交给本班学习委员（如 201711111111_蔡一_lab1.zip）。
- 学习委员将本班代码压缩打包，于 10 月 17 日下午 2 点前发送到我的邮箱。