

实验四 存储管理

实验目的

在 OpenEuler / Linux 操作系统上,通过模拟实现按需调页式存储管理的几种基本页面置换算法,了解虚拟存储技术的特点,掌握虚拟存储按需调页式存储管理中几种基本页面置换算法的基本思想和实现过程,并比较它们的效率。

实验内容

1. 生成内存访问串

首先用 `srand()` 和 `rand()` 函数定义和产生指令地址序列,然后将指令地址序列转换成相应的页地址流。

比如:通过随机数产生一个内存地址,共 **100 (可以自己定义)** 个地址,地址按下述原则生成:

- (1) 70%的指令是顺序执行的
- (2) 10%的指令是均匀分布在前地址部分
- (3) 20%的指令是均匀分布在后地址部分

具体的实施方法是:

- a) 从地址 0 开始;
- b) 若当前指令地址为 m ,按上面的概率确定要执行的下一条指令地址,分别为顺序、在前和在后:
 - 顺序执行:地址为 $m+1$ 的指令;
 - 在前地址: $[0, m-1]$ 中依前面说明的概率随机选取地址;
 - 在后地址: $[m+1, 99]$ 中依前面说明的概率随机选取地址;
- c) 重复 b) 直至生成 100 个指令地址。

假设每个页面可以存放 **10 (可以自己定义)** 条指令,将指令地址映射到页面,生成内存访问串。

2. 设计并实现下述算法,计算访问缺页率,并对算法的性能加以比较。

- (1) 最优置换算法 (Optimal)
 - (2) 最近最少使用 (Least Recently Used)
 - (3) 先进先出法 (First In First Out)
- 其中,缺页率 = 页面失效次数 / 页地址流长度

要求:

分析在同样的内存访问串上执行,分配的物理内存块数量和缺页率之间的关系;并在同样情况下,对不同置换算法的缺页率比较。