
请求调页存储管理方式模拟

操作系统课程设计

OPERATING SYSTEM, SPRING 2017

By

1552674 李源

Tongji University
School of Software Engineering

Contents

1	项目背景	3
1.1	项目需求	3
1.2	项目目的	3
2	需求分析	3
2.1	模拟方案	3
3	调度算法	4
3.1	置换算法	4
4	系统实现	4
5	开发环境	4
6	提交内容	4

1 项目背景

1.1 项目需求

编写一个应用程序，实现请求调页存储管理方式模拟。假设每个页面可存放10条指令，分配给一个作业的内存块为4。模拟一个作业的执行过程，该作业有320条指令，即它的地址空间为32页，目前所有页还没有调入内存。采用FIFO算法或LRU算法实现置换。指令分布应该是均匀的。

1.2 项目目的

- (1) 掌握页面、页表、地址转换过程；
- (2) 对页面置换过程有更深入的认识；
- (3) 加深对请求调页系统的原理和实现过程的理解。

2 需求分析

根据项目需求，我们可以得知本项目所模拟的内存和作业分别满足如下要求：

- (1) 内存：4个内存块，一个内存块中能存放10条指令；
- (2) 作业：320条指令，分别放在32页中；需执行320条指令，且指令可以重复。
- (3) 指令分布：50%的指令是顺序执行的，25%是均匀分布在前地址部分，25%是均匀分布在后地址部分。
- (4) 置换算法：选择LRU算法。

结合内存和作业的实际情况，我们可以设计出本项目的模拟方案。

2.1 模拟方案

该模拟方案的步骤，大致可按照如下进行：

- (1) 随机从一条指令开始执行。
- (2) 如果指令在内存中，则显示其物理地址，并转到下一条指令；如果不在内存中，则发生缺页，此时需要记录缺页次数，并将其调入内存。
- (3) 如此循环，直到执行了320条指令，之后不再执行。

3 调度算法

根据前文对本模拟系统的分析，以及对两种指令情况的具体考虑，可以将该调度算法细分为两个部分：

- (1) 实现内存块的置换算法，即LRU算法；
- (2) 实现指令的随机选择算法，可称为random算法。

3.1 置换算法

根据LRU算法的定义，使用离过去最近作为不远将来的近似，将最长时间没有使用的页置换出去。则LRU算法的关键是，选择最长时间没有使用的页，并且置换出去。在本模拟系统中，我使用一个数组完成LRU算法中对于最长时间没有使用的页寻找。

- (1) 该数组存储四个内存块各自上次使用的时间。
- (2) 初始化的时候，该数组所有的值被初始化为0。
- (3) 每次执行指令时，首先检测该指令是否在某一个内存块中。
- (4)

4 系统实现

5 开发环境

- 系统： macOS Sierra (version 10.12.4)
- IDE： Qt Creator 4.2.1, Based on Qt 5.8.0 (Clang 7.0 (Apple), 64 bit)
- 语言： C++

6 提交内容

- 源代码
- assignment2.zip 可执行文件压缩包（需在mac系统下使用）
- assignment2.dmg 安装包（需在mac系统下使用）
- 说明文档

- 演示视频