《操作系统》实验报告

[Linux]虚存管理

|  |  |  |
| --- | --- | --- |
| 实验小组（24组） | 学号 | 姓名 |
| 实验负责人 | 14061219 | 刘朋涛 |
| 小组成员 | 14061182 | 柴瑞 |
| 小组成员 | 14061220 | 熊藤攀 |
| 小组成员 | 14231028 | 金银株 |

**1.实验要求说明**

## 基本需求

通过本实验，要求学生能够了解Linux系统下页式存储管理机制，并实现一个简单的虚存管理模拟程序。具体要求如下：

（1）设计并实现一个虚存管理模拟程序，模拟一个单道程序的页式存储管理，用一个一维数组模拟实存空间，用一个文本文件模拟辅存空间。

（2）建立一个一级页表。

（3）程序中使用一个函数do\_request()随机产生访存请求，访存操作包括读取、写入、执行三种类型。

（4）实现一个函数do\_response()响应访存请求，完成虚地址到实地址的定位及读/写/执行操作，同时判断并处理缺页中断。

（5）实现LFU页面淘汰算法。

## 进阶需求

要求学生在完成上述基本要求基础上对程序的功能和性能进行改进，改进建议如下：

（1）实现多道程序的存储控制。

（2）建立一个多级页表或快表。

（3）将do\_request()和do\_response()实现在不同进程中，通过进程间通信（如FIFO）完成访存控制的模拟。

（4）实现其它页面淘汰算法：先进先出（FIFO）、最近最久未使用淘汰算法（LRU）、最不频繁使用淘汰算法（LFU）、最优算法（OPT）。

# 设计说明

## 结构设计



图2.1 程序结构设计图

1. **实验要求完成情况**

## 3.1实现了所有要求的基本功能和建议的扩展功能

## 3.2关键功能具体说明

### 多道程序的体现

在此，我们假设我们的虚存文件里共存有两个程序，两个程序均分了虚存的大小，我们的续存共有64\*4，页面大小为4，续存共有64页，每个程序占了32页大小的空间。我们需要对每一个程序建立其对应的页表，我们采用的是两级页表的形式，对于每个程序来说一级页表共需8个，为了简化操作，我们将一级页表连在了一起，故一级页表的标识为0~15，另外建立了一个结构，表征对于每个程序，它的一级起始与结束分别是什么。这样，得到一个逻辑地址后，我们就知道其随机产生的请求是针对哪个程序进行的。

### 多级页表的实现

一级页表所需的地址空间必须是内存中一片连续的区域，采用二级页表的话，一级必须连续，映射到得二级页表可以不是连续的，这样更优于内存的使用。我们的虚存共有64页，这是二级页表需要的大小，我们再将二级页表项的每四个分为一组，由一个一级页表指向，这样一级页表共有16个，各级页表都是由一结构体现，一级较为简单，只需对应到二级页表的映像，二级页表需要存放各类信息。

### 进程间通信的实现

进程间通信采用的是FIFO机制，通过FIFO实现do\_request与do\_response之间的通信，将do\_request写在一简单程序里，为vm\_fifo.c，在vm\_fifo.c里打开FIFO，按照用户的需求，产生一定数量的请求，并将产生的请求写入FIFO。响应程序do\_response()在vm\_simulator.c，通过在vm\_simulator.c里面打开FIFO，读出里面的数据进行解析，再进一步响应，这样就可以实现进程间的信息交流。

### 页面替换算法

**FIFO算法的实现**：我们定义了一个数组Time，用来存放页面装入的先后次序，Time[0]总是存放最先放入的页面，每次需要替换页面时，只需替换Time[0]即可，替换后，将Time中的数据整体前移一位，把替换掉的页面记录冲掉。

**LRU算法的实现**：全局变量time记录了当前所有页面所执行的次数，二级页表的结构中设有unsigned int型的变量usetime，把当前的time赋给usetime，表示这个页面在第time次被执行，因此，这个值越小表明越久没有用到，因为每次用到都会刷新usetime的值，所以，每次需要替换页面时，找usetime最小的页面替换。

**收获总结**

对虚存管理机制有了进一步的了解，包括地址转换，页面装入，页面替换，存储控制等问题有了更加深刻的体会。

深入的了解了其中某些算法的实现，在对虚存管理的模拟中将学到的理论加以练习和理解。