



**操作系统课程设计实验报告**

——实验三：虚存管理

负责人姓名：邓修

学号：14061170

日期：2016.5.25

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **学号** | **实验分工** |
| **1** | 杨汀阳 | 14061164 | 实验一 |
| **2** | 王柏琦 | 14061158 | 实验二 |
| **3** | 邓修 | 14061170 | 实验三 |
| **4** | 王蔡勐 | 14061152 | 实验四 |

目录

[1 实验目的 4](#_Toc451023347)

[2 需求说明 4](#_Toc451023348)

[2.1 基本要求 4](#_Toc451023351)

[2.2 提高要求 4](#_Toc451023352)

[2.3 完成情况 5](#_Toc451023353)

[3 设计说明 6](#_Toc451023354)

[3.1 流程图 6](#_Toc451023358)

[3.2 基本要求实现说明 7](#_Toc451023359)

[3.1.1 模拟存储 7](#_Toc451023364)

[3.1.2 一级页表 7](#_Toc451023365)

[3.1.3 产生访存请求 7](#_Toc451023366)

[3.1.4 响应请求 7](#_Toc451023367)

[3.2 提高要求实现说明 7](#_Toc451023368)

[3.2.1 实现三级页表 7](#_Toc451023369)

[3.2.2 多道程序的存取控制 1](#_Toc451023370)2

[3.2.3 页面老化算法 13](#_Toc451023372)

[4 收获和感想 16](#_Toc451023383)

1. 实验目的
   1. 了解Linux的内存管理机制。
   2. 掌握页式虚拟存储技术，理解虚地址到实地址的定位过程。
   3. 掌握“最不频繁使用淘汰算法”，即LFU页面淘汰算法。
2. 需求说明
4. 1. 基本要求

本实验要求在了解Linux系统下页式存储管理机制的基础下，实现一个简单的虚存管理模拟程序。

具体要求如下：

* 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。
* 建立一个一级页表
* 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
* 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
* 实现LFU页面淘汰算法
  1. 提高要求

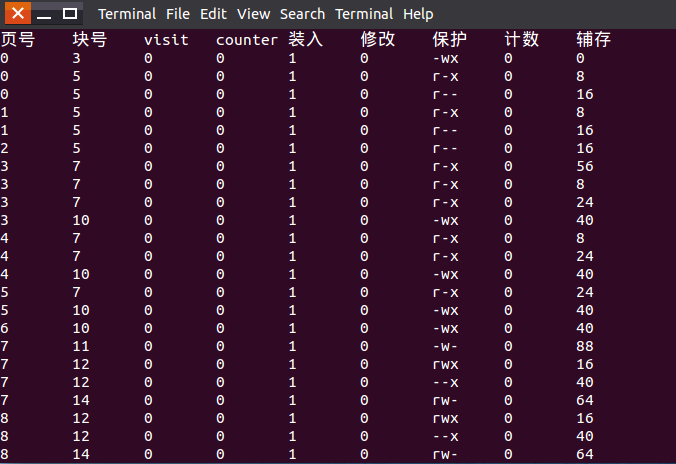
在基础要求完成的基础上对程序的功能和性能进行改进，具体要求如下：

* 建立一个多级页表
* 实现多道程序的存储控制
* 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
* 实现其它页面淘汰算法：如页面老化算法、最近最久未使用淘汰算法（LRU）、最优算法（OPT）等
  1. 完成情况

所有基本要求已经完成，提高要求的完成情况如下：

* 建立了一个的三级页表。
* 通过修改请求的数据结构和do\_response()等函数，实现了多道程序的存取控制。
* 实现了页面老化算法。

程序运行截图如下：



1. 设计说明

4. 1. 流程图

结束

开始

更新页表

装入页面

产生缺页中断

获取块号，计算实地址

写回至辅存

产生访存请求，写入管道

获取页表

根据页号和页内偏移查页表

更新页表

返回访存结果

返回错误信息

淘汰页面

环境初始化

地址越界

页在实存

有权访问

页有修改

有空闲块

程序结束

是

是

是

是

是

是

否

否

否

否

否

否

* 1. 基本要求实现说明

源程序基本实现了所有基本要求，说明如下:

3. 1. 1. 模拟存储

实验用一维数组模拟实存空间，用文本文件模拟辅存空间。通过文件操作实现实存与辅存中的数据读取与写入。

* + 1. 一级页表

实验中采用一维数组作为页表项，计算页号后通过数组下标查找页表。

* + 1. 产生访存请求

do\_request()函数通过随机数方式产生访存请求的类型、虚地址和进程号，若访存请求为WRITE，则还需产生一个待写入的值，并将所有内容填入一个MemoryAccessRequest结构体中，交给do\_response()函数进行访存操作。

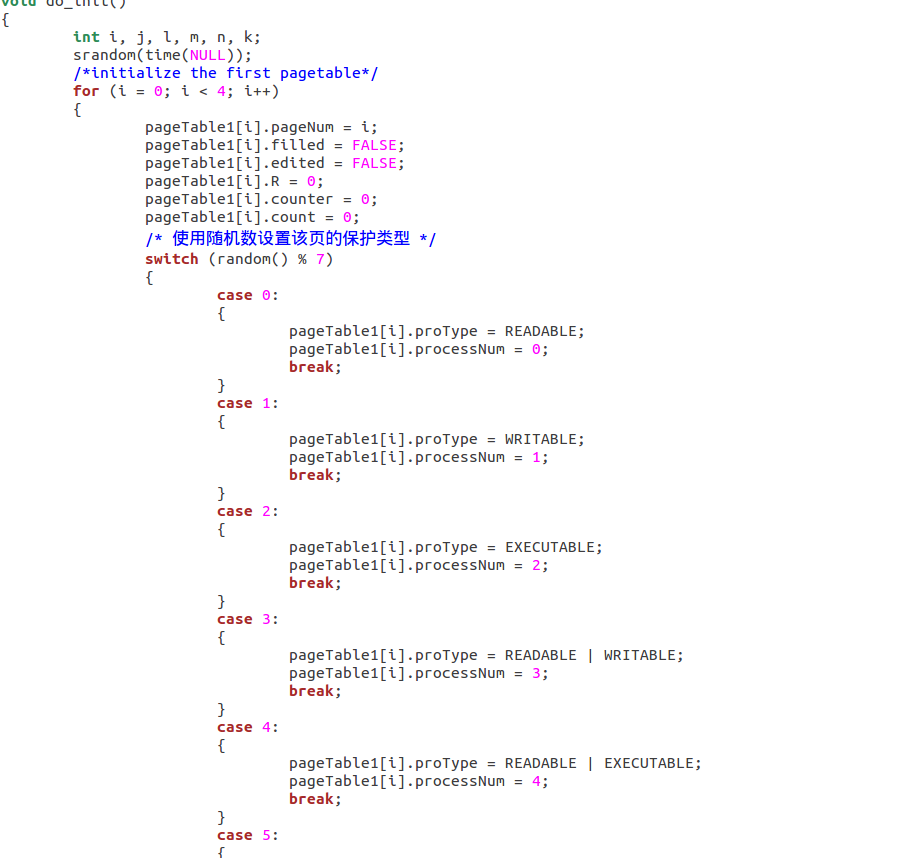
* + 1. 响应请求

先根据请求的地址计算页号和页内偏移值，获取对应页表项。再看该页面是否已装入内存，若没有装入则产生缺页中断，将页面调入内存。然后检查页面的访问权限，判断该请求是否可以访问。若不能访问则输出错误信息，否则响应请求。

* 1. 提高要求实现说明
     1. 实现三级页表

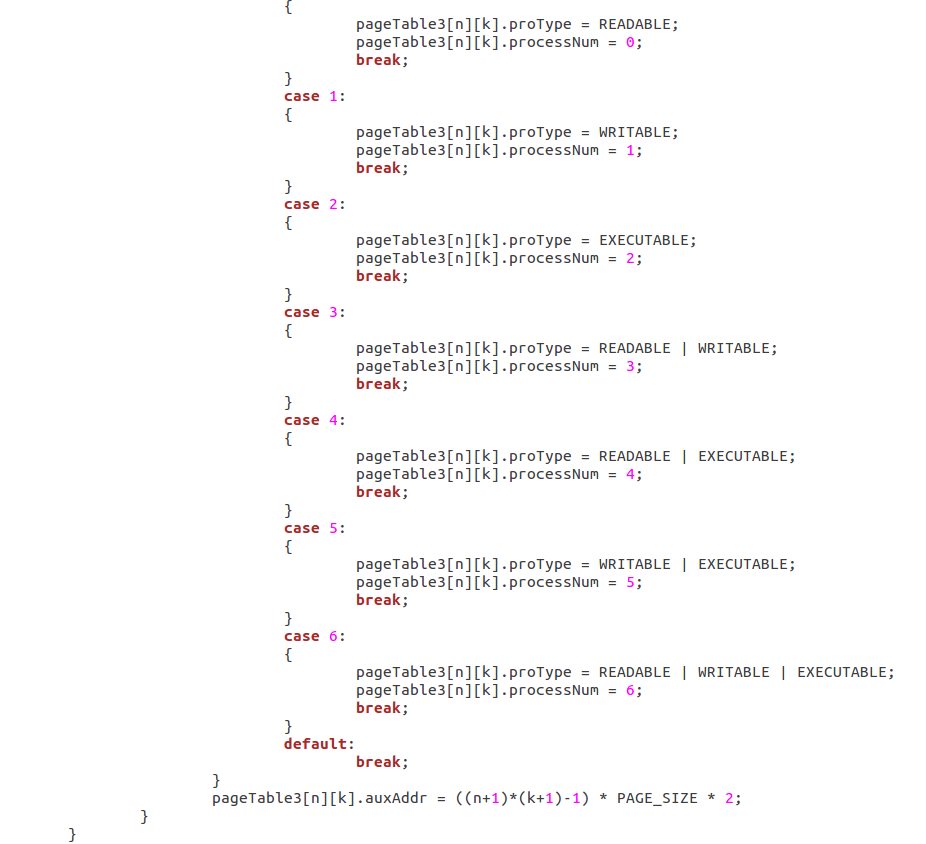
实现思路：本实验中将原来的64个二级页表分为16份，每四个为一个三级页表，用4个二级页表指向相应三级页表地址。实现代码如下。

初始化：









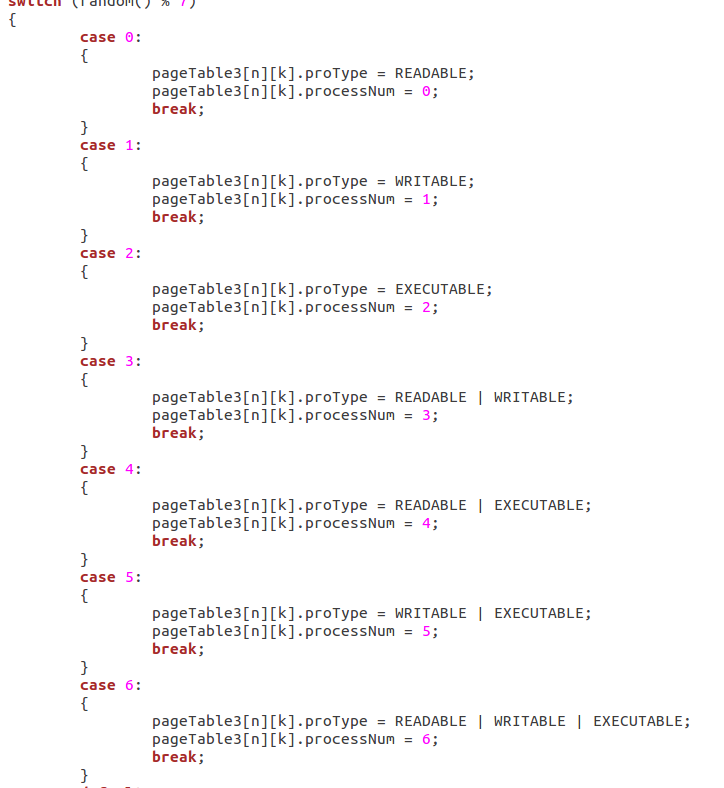
随机选择一些物理块进行页面装入：



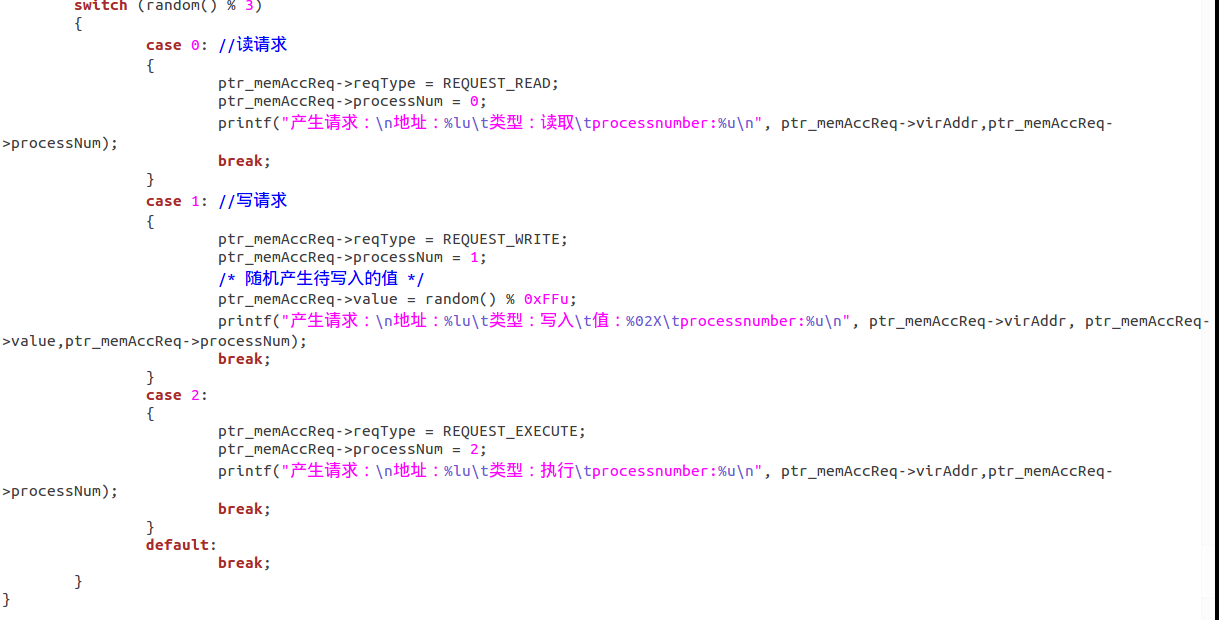
* + 1. 多道程序的存取控制

本次实验中实现的是3道进程的存取控制。实现思路为：为每个页表设置属于哪些进程，产生请求时标记该请求所属的进程。程序处理请求时先检查访问的权限，若不能访问则输出提示信息，能访问则继续处理。具体实现过程如下。

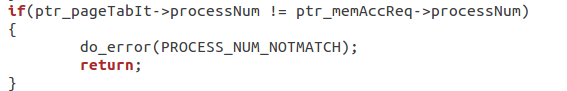
1. 随机设置页表属于哪些进程：



2.在产生请求时标记该请求属于哪个进程



3.在访问内存时进行权限判断



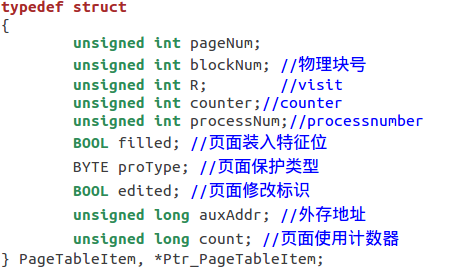
* + 1. 页面老化算法

1. 算法的基本思想与实现方法介绍

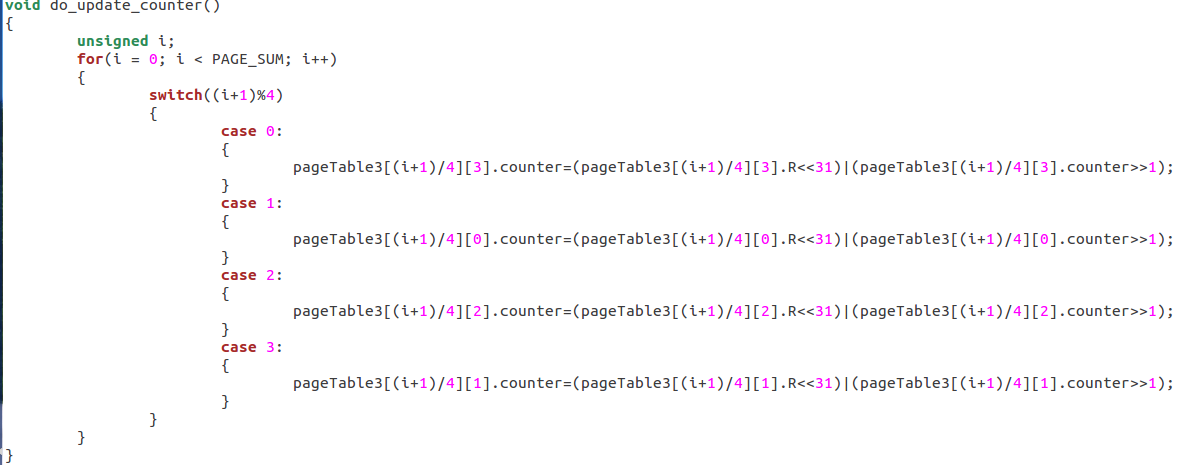
页面老化算法是LRU的一种近似算法，计数过程与时钟相关。本次实验采用的实现方法是每个页表项中添加一个计数器和一个访问位R。每次访问页面时，将被访问的页表项的R置为1。每过一个时间段更新页表项的计数器：先将计数器右移一位，再将R位加到计数器的最左端的位。当需要替换页面时，选择内存中计数值最小的页面替换。

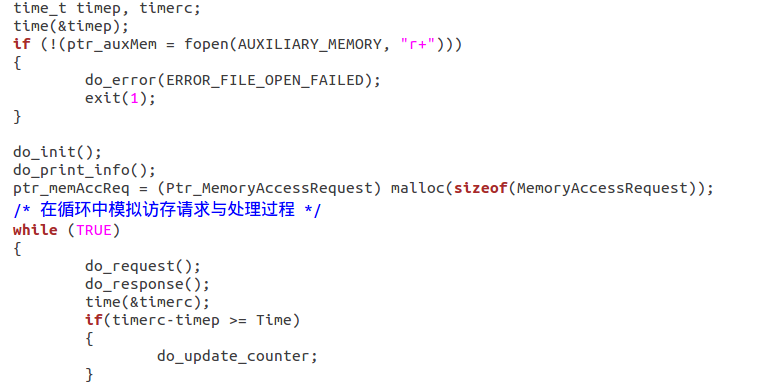
1. 实现代码

在页表项结构中添加计数器和访问位R：



每过一定时间段更新计数器：





缺页中断时调用do\_PAA算法选择计数值最小的页面替换：



1. 收获和感想

经过这次实验，我们熟悉了Linux系统下的页式存储管理机制，在实验中我们对课上所学的虚存管理的相关理论知识的有了更深层次的理解，还间接巩固了进程通信的知识。在查找资料的过程中对页面替换算法有了更多的理解。同时，在编写程序的过程中，提高了我们分析问题、发现并解决问题的能力以及团队成员间的沟通协作能力。

本次实验并不太难，但我们在实验过程中也遇到了一些困难。首先是对一些实验要求有不同理解，各组员提出了不同实现方案。其次是编程时遇到的问题，如使用未初始化的指针等。不过我们通过上网查询相关资料、与同学们讨论、询问助教、使用gdb调试等方式逐一解决了这些问题。正是在不断发现和解决问题的过程中，我们不断得到了提高。