

苏州大学预习报告

院、系	计算机学院	年级专业		姓名		学号	
课程名称	操作系统课程实践					成绩	
指导教师	褚晓敏	同组实验者	无	实验日期			

实验名称

第8章 虚拟内存管理

一、实验目的

- 1、学习操作系统的虚拟内存管理原理。
- 2、理解操作系统内存管理的分页、虚拟内存、“按需调页”思想及方法。
- 3、掌握 Linux 内核对虚拟内存、虚存段、分页式内存管理、按需调页的实现机制。

二、实验内容

- (1) 统计自内核加载完成以后到当前时刻为止发生的缺页次数和经历过的时间。
- (2) 统计从当前时刻起一段时间内发生地缺页中断次数。

三、知识储备（参考课本）

1、Linux 虚拟内存管理

(1) 虚拟内存的抽象模型

- 1) 换页
- 2) 交换

被修改过的 dirty 页面换出时需保存在交换空间。

(2) Linux 的分页管理

三级页表->四级页表

Intel x86 体系中，Linux 实际使用 2 级页表

分析了 `/include/asm-i386/pgtable-2level-defs.h` 和 `include/linux/sched.h` 中的部分代码

(3) 虚存段 (vma) 的组织和管理

程序执行时，可执行文件只是连接到进程的虚拟内存。当程序的其他部分运行需要引用到这部分时才把它们从磁盘调入内存。

进程可用的虚存空间共 4G，0-3G 可直接使用，剩下 1G 属于内核。在创建用户进程时，内核的代码段和数据段被映射到虚拟内存地址 3GB 以上的虚存空间，供内核态进程使用。（事实上，所有进程的 3GB-4GB 的虚存空间的映像）

一个进程在运行过程中使用的物理内存一般是不连续的，用到的虚拟地址也不是连成一片的，而是被分成几块，进程通常占用几个虚存段，分别用于代码段、数据段、堆栈段等。每个进程的所有虚存段通过指针构成链表，虚存段在此链表中的排列顺序按照他们的地址增长顺序进行。为了便于理解，Linux 定义了虚存段 vma。一个 vma 段是属于某个进程的一段连续的虚存空间，在

这段虚存里所有页面拥有一些相同的特征。例如，属于同一进程，相同权限等。

教材分析了 `include/linux/mm.h` 和 `include/linux/rbtree.h` 等代码。

（4）页面分配与回收

1) 页面分配

2) 页面回收

3) 按需调页

详细分析了缺页中断处理函数 `do_page_fault`，定义在 `arch/i386/mm/fault.c` 中。

后续还分析了其中调用的一些函数，如

`mm/memory.c` 中的 `handle_pte_fault` 函数、`do_no_page` 函数、`do_swap_page` 函数等。

四、实验大致流程

1、统计系统启动以来的缺页次数和时间

（1）实验原理

发生一次中断就要进入缺页中断服务函数 `do_page_fault` 一次，所以可以定义一个全局变量 `pfcount` 作为计数变量，在执行 `do_page_fault` 时，该变量值加 1。该值即为缺页次数。

系统启动以来的时间，可以利用系统原有的变量 `jiffies`，这是一个系统的计时器，从内核加载完开始计时，以 10ms 为计时单位。

读取 `pfcount` 和 `jiffies` 需要借助 `/proc` 文件系统，需要在 `/proc` 文件系统下建立 `pf` 目录，并在目录下建立文件 `pfcount` 和 `jiffies`。

（2）实验步骤

- 1) 在 `include/linux/mm.h` 中声明变量 `pfcount`: `extern unsigned long pfcount;`
- 2) 在 `arch/i386/mm/fault.c` 文件中定义变量 `pfcount`，并在 `do_page_fault()` 内将 `pfcount++`。
- 3) 在 `kernel/time.c` 中声明 `pfcount`，并加入 `EXPORT_SYMBOL(pfcount)`，使内核模块能读取变量 `pfcount`。
- 4) 重新编译内核。
- 5) 准备一个内核模块和其 `Makefile`，用于在 `/proc` 中读取 `pfcount` 和 `jiffies` 变量。
- 6) 编译完成后没几分钟运行一次，比较缺页次数和时间。
(也不知道能不能成功)

2、统计一段时间内系统缺页次数

（1）实验原理

在 Linux 的 `/proc` 文件系统中有一个记录当前基本状况的文件 `stst`。该文件中有一节是关于中断次数的。这一节以 `intr` 开头，后面存储着各中断发生的次数。缺页中断是第 14 号，也就是 `intr` 后的第 16 项。所以该实验可以通过在当前时刻和实验结束时刻分别读取缺页中断的次数，通过简单的减法就能得到这段时间内缺页中断的次数。

（2）实验步骤

编写程序文件 `pfintr.c`。(代码教材上有，基本就是读文件配合定时器)