

# 浙江大学实验报告

课程名称： 操作系统 实验类型： 综合型/设计性

实验项目名称： 实验二 添加系统调用

学生姓名： 钱旭峰 专业： 计算机科学 学号： 3140102491

电子邮件地址： 17816862315@163.com 手机： 17816862315

实验日期： 2016 年 12 月 30 日

## 一、实验目的

学习重建 Linux 内核。

学习 Linux 内核的系统调用，理解、掌握 Linux 系统调用的实现框架、用户界面、参数传递、进入/返回过程。阅读 Linux 内核源代码，通过添加一个简单的系统调用实验，进一步理解 Linux 操作系统处理系统调用的统一流程。了解 Linux 操作系统缺页处理，进一步掌握 task\_struct 结构的作用。

## 二、实验内容

在现有的系统中添加一个不用传递参数的系统调用。这个系统调用的功能是实现统计操作系统缺页总次数，当前进程的缺页次数，以及每个进程的“脏”页面数。严格来说这里讲的“缺页次数”实际上是页错误次数，即调用 do\_page\_fault 函数的次数。实验主要内容：

- 在 Linux 操作系统环境下重建内核
- 添加系统调用的名字
- 利用标准 C 库进行包装
- 添加系统调用号
- 在系统调用表中添加相应表项
- 修改统计缺页次数、“脏”页相关的内核结构和函数
- sys\_mysyscall 的实现
- 编写用户态测试程序

## 三、主要仪器设备（必填）

华硕 X450V，unbantu 操作系统，虚拟机 2G 内存

## 四、操作方法和实验步骤

1. 首先先把 linux4.6.0 的内核解压到指定目录

2. 安装 libncurses5-dev

3. 清除目录下所有配置文件和先前生成核心时产生的.o 文件，为了与正在运行的操作系统内核的运行环境匹配，可以先把当前已配置好的文件复制到当前目录下，新的文件名为.config 文件

4. 添加系统调用号，内容与指导书上的一致，唯一的不同在于：

```
arch/x86/entry/syscalls/syscall_64.tbl)
```

---

```
223      common  mysyscall      sys_mysyscall
```

---

5. 修改统计系统缺页次数和进程缺页次数的内核代码（与指导书上的一致）

6. sys\_mysyscall 的实现（包括缺页和脏页的统计）：

```
//new add stw
pte_t* get_page_point(struct mm_struct* mm, unsigned long virt)
{
    pgd_t* pgd = pgd_offset(mm, virt);
    if (pgd_none(*pgd) || pgd_bad(*pgd))
        return NULL;
    pud_t* pud = pud_offset(pgd, virt);
    if (pud_none(*pud) || pud_bad(*pud))
        return NULL;
    pmd_t* pmd = pmd_offset(pud, virt);
    if (pmd_none(*pmd) || pmd_bad(*pmd))
        return NULL;
    pte_t* pte = pte_offset_map(pmd, virt);
    if (!pte) return NULL;
    return pte;
}
```

```

//new add stw

asmlinkage int sys_mysyscall(void)
{
    struct task_struct *p;
    for (p = &init_task; next_task(p) != &init_task; p = next_task(p))
    {
        if (p->mm && p->mm->mmap)
        {
            struct mm_struct *mm = p->mm;
            struct vm_area_struct *vma;
            unsigned long vpage;
            int dirty_page = 0;
            for (vma = mm->mmap; vma; vma = vma->vm_next)
            for (vpage = vma->vm_start; vpage < vma->vm_end; vpage += PAGE_SIZE)
            {
                pte_t* pte = get_page_point(mm, vpage);
                if (pte)
                {
                    if (pte_dirty(*pte)) dirty_page++;
                }
            }

            printk("$Pid:%d    $Dirty page(1):%d    $Dirty page(2):%d    $Page
faults:%lu\n", p->pid, dirty_page, p->nr_dirtied, p->pf);
        }
        else
        {
            printk("$Pid:%d    $Dirty page(1):0    $Dirty page(2):%d    $Page
faults:%lu\n", p->pid, p->nr_dirtied, p->pf);
        }
    }

    printk("Total page faults: %lu\n", pfcount);
}

```

```

    printk("Current process pid: %d\n", current->pid);

    printk("Page faults in current process: %lu\n", current->pf);

    return 0;
}

```

7.编译内核和重启内核

8.编写用户态程序:

```

#include <linux/unistd.h>

#include <sys/syscall.h>

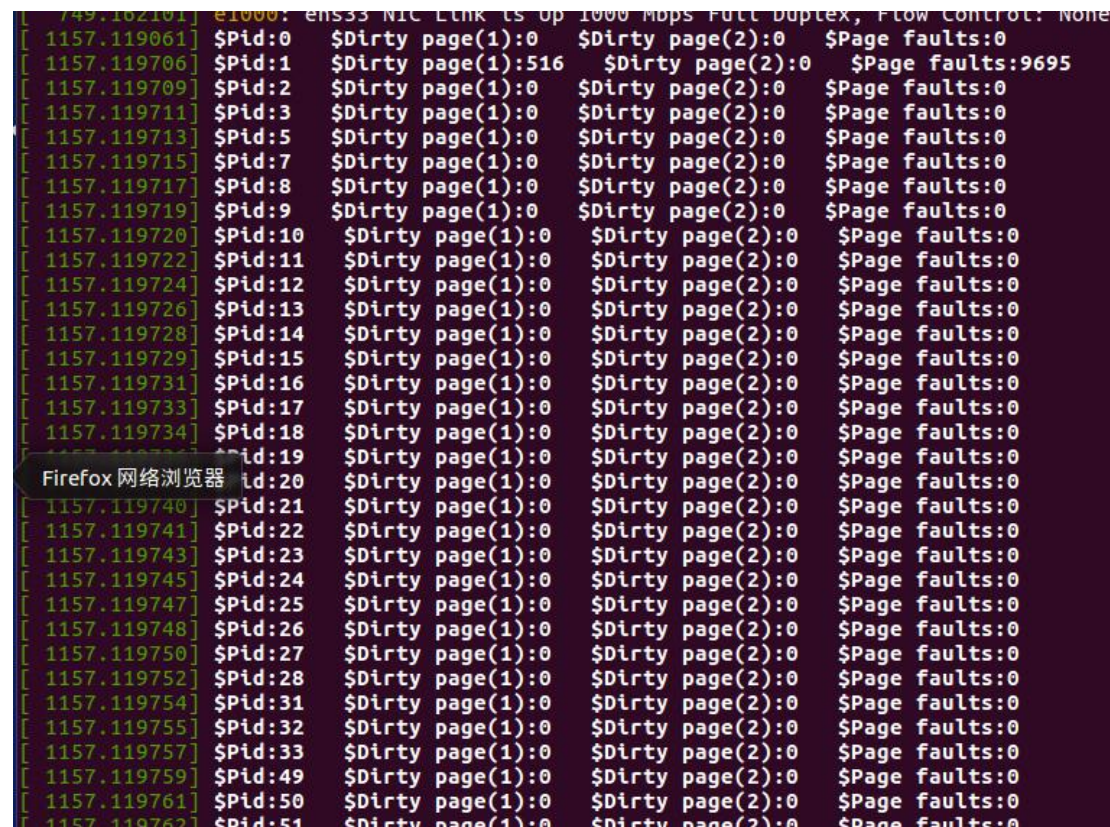
#define __NR_mysyscall 223

int main()
{
    syscall(__NR_mysyscall);

    return 0;
}

```

## 五、实验结果和分析



```

[ 749.162101] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 1157.119061] $Pid:0 $Dirty page(1):0 $Dirty page(2):0 $Page faults:9695
[ 1157.119706] $Pid:1 $Dirty page(1):516 $Dirty page(2):0 $Page faults:0
[ 1157.119709] $Pid:2 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119711] $Pid:3 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119713] $Pid:5 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119715] $Pid:7 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119717] $Pid:8 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119719] $Pid:9 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119720] $Pid:10 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119722] $Pid:11 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119724] $Pid:12 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119726] $Pid:13 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119728] $Pid:14 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119729] $Pid:15 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119731] $Pid:16 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119733] $Pid:17 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119734] $Pid:18 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119735] $Pid:19 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119740] $Pid:21 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119741] $Pid:22 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119743] $Pid:23 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119745] $Pid:24 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119747] $Pid:25 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119748] $Pid:26 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119750] $Pid:27 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119752] $Pid:28 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119754] $Pid:31 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119755] $Pid:32 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119757] $Pid:33 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119759] $Pid:49 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119761] $Pid:50 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119762] $Pid:51 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0

```



```

[ 1157.119936] $Pid:269 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119938] $Pid:271 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119940] $Pid:306 $Dirty page(1):0 $Dirty page(2):784 $Page faults:0
[ 1157.119942] $Pid:307 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.119943] $Pid:334 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.120187] $Pid:338 $Dirty page(1):330 $Dirty page(2):0 $Page faults:2577
[ 1157.120519] $Pid:358 $Dirty page(1):742 $Dirty page(2):0 $Page faults:6011
[ 1157.120522] $Pid:413 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.120524] $Pid:560 $Dirty page(1):0 $Dirty page(2):4 $Page faults:0
[ 1157.120526] $Pid:561 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.121548] $Pid:762 $Dirty page(1):172 $Dirty page(2):0 $Page faults:4752
[ 1157.122604] $Pid:791 $Dirty page(1):134 $Dirty page(2):0 $Page faults:4812
[ 1157.124000] $Pid:852 $Dirty page(1):617 $Dirty page(2):19 $Page faults:6629
[ 1157.124662] $Pid:933 $Dirty page(1):725 $Dirty page(2):5 $Page faults:6272
[ 1157.126798] $Pid:992 $Dirty page(1):231 $Dirty page(2):28 $Page faults:5446
[ 1157.127065] $Pid:994 $Dirty page(1):52 $Dirty page(2):0 $Page faults:5405
[ 1157.128430] $Pid:997 $Dirty page(1):208 $Dirty page(2):0 $Page faults:5907
[ 1157.128488] $Pid:1003 $Dirty page(1):45 $Dirty page(2):0 $Page faults:5545
[ 1157.128702] $Pid:1004 $Dirty page(1):71 $Dirty page(2):0 $Page faults:5664
[ 1157.130884] $Pid:1005 $Dirty page(1):434 $Dirty page(2):1 $Page faults:6329
[ 1157.131153] $Pid:1009 $Dirty page(1):412 $Dirty page(2):0 $Page faults:6476
[ 1157.132928] $Pid:1025 $Dirty page(1):430 $Dirty page(2):0 $Page faults:6512
[ 1157.132928] $Pid:1027 $Dirty page(1):241 $Dirty page(2):0 $Page faults:6162
[ 1157.132928] $Pid:1029 $Dirty page(1):106 $Dirty page(2):0 $Page faults:6240
[ 1157.135265] $Pid:1030 $Dirty page(1):184 $Dirty page(2):0 $Page faults:6453
[ 1157.137222] $Pid:1031 $Dirty page(1):662 $Dirty page(2):8 $Page faults:8874
[ 1157.137570] $Pid:1037 $Dirty page(1):98 $Dirty page(2):0 $Page faults:6508
[ 1157.137842] $Pid:1097 $Dirty page(1):83 $Dirty page(2):0 $Page faults:6490
[ 1157.139307] $Pid:1104 $Dirty page(1):335 $Dirty page(2):0 $Page faults:7302
[ 1157.140819] $Pid:1108 $Dirty page(1):669 $Dirty page(2):0 $Page faults:8978
[ 1157.142561] $Pid:1159 $Dirty page(1):207 $Dirty page(2):27 $Page faults:7677
[ 1157.142970] $Pid:1164 $Dirty page(1):179 $Dirty page(2):0 $Page faults:9043
[ 1157.146260] $Pid:1197 $Dirty page(1):5292 $Dirty page(2):27 $Page faults:31525
[ 1157.146260] $Pid:1216 $Dirty page(1):88 $Dirty page(2):0 $Page faults:7626

系统设置
[ 1157.200240] $Pid:2236 $Dirty page(1):277 $Dirty page(2):0 $Page faults:12903
[ 1157.200913] $Pid:2262 $Dirty page(1):241 $Dirty page(2):0 $Page faults:12145
[ 1157.201269] $Pid:2273 $Dirty page(1):146 $Dirty page(2):13 $Page faults:12884
[ 1157.203253] $Pid:2282 $Dirty page(1):1028 $Dirty page(2):0 $Page faults:31255
[ 1157.204825] $Pid:2296 $Dirty page(1):1051 $Dirty page(2):0 $Page faults:31329
[ 1157.206153] $Pid:2314 $Dirty page(1):738 $Dirty page(2):0 $Page faults:13705
[ 1157.207240] $Pid:2316 $Dirty page(1):564 $Dirty page(2):0 $Page faults:16527
[ 1157.207254] $Pid:2323 $Dirty page(1):25 $Dirty page(2):0 $Page faults:12964
[ 1157.207979] $Pid:2330 $Dirty page(1):284 $Dirty page(2):48 $Page faults:13507
[ 1157.209591] $Pid:2333 $Dirty page(1):1294 $Dirty page(2):7 $Page faults:15321
[ 1157.210161] $Pid:2346 $Dirty page(1):1069 $Dirty page(2):4 $Page faults:14471
[ 1157.210162] $Pid:2392 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.210162] LibreOffice Writer $Pid:2425 $Dirty page(1):808 $Dirty page(2):0 $Page faults:18240
[ 1157.212415] $Pid:2455 $Dirty page(1):262 $Dirty page(2):0 $Page faults:16238
[ 1157.213820] $Pid:2492 $Dirty page(1):1354 $Dirty page(2):0 $Page faults:17261
[ 1157.213890] $Pid:2499 $Dirty page(1):441 $Dirty page(2):0 $Page faults:17311
[ 1157.213891] $Pid:2521 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.215028] $Pid:2598 $Dirty page(1):1077 $Dirty page(2):0 $Page faults:16774
[ 1157.215070] $Pid:2796 $Dirty page(1):215 $Dirty page(2):1 $Page faults:8991
[ 1157.215170] $Pid:2968 $Dirty page(1):107 $Dirty page(2):0 $Page faults:18675
[ 1157.215274] $Pid:2969 $Dirty page(1):102 $Dirty page(2):0 $Page faults:18911
[ 1157.215614] $Pid:2972 $Dirty page(1):167 $Dirty page(2):30 $Page faults:19171
[ 1157.215615] $Pid:2984 $Dirty page(1):0 $Dirty page(2):0 $Page faults:0
[ 1157.216240] $Pid:2998 $Dirty page(1):236 $Dirty page(2):0 $Page faults:12249
[ 1157.218443] $Pid:3025 $Dirty page(1):32951 $Dirty page(2):83 $Page faults:91135
[ 1157.219128] $Pid:3087 $Dirty page(1):232 $Dirty page(2):0 $Page faults:12484
[ 1157.219169] $Pid:3117 $Dirty page(1):47 $Dirty page(2):0 $Page faults:9817
[ 1157.219170] Total page faults: 776557
[ 1157.219171] Current process pid: 3119
[ 1157.219171] Page faults in current process: 17368
anthony@anthony-virtual-machine:~$

```

实验正确输出了缺页和当前进程缺页和每个进程的脏页数

## 六、问题解答

1. 多次运行 test 程序，每次运行 test 后记录下系统缺页次数和当

前进程缺页次数，给出这些数据。`test` 程序打印的缺页次数是否就是操作系统原理上的缺页次数？

答：`test` 程序打印的缺页次数不是操作系统原理上的缺页次数。操作系统原理上的缺页次数是指当内存中不存在指定页面而发生缺页中断的次数。实验中指的是调用 `do_page_fault()` 函数的次数。

2. 除了通过修改内核来添加一个系统调用外，还有其他的添加或修改一个系统调用的方法吗？如果有，请论述。

答：添加系统调用的第二种方法：内核模块法。这种方法是采用系统调用拦截的一种方式，改变某一个系统调用号对应的服务程序为我们自己的编写的程序，从而相当于添加了我们自己的系统调用。

3. 对于一个操作系统而言，你认为修改系统调用的方法安全吗？请发表你的观点。答：我认为不安全，如果能随意的修改系统调用，就有可能误将操作系统内核的文件修改掉，会导致系统奔溃。

## 七、讨论、心得

我在缺页的统计上一开始遇到了许多问题，之后一点一点的查资料终于搞定了缺页。这个实验是我对系统调用与 linux 内核有了重新的认识，是我进一步了解了如何添加系统调用。