添加系统调用

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。

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
- 修改统计缺页次数相关的内核结构和函数
- sys_mysyscall的实现
- 编写用户态测试程序

实验指导

下面的指导是以ubuntu 16.04（64位）和kernel 4.8为例，不同的Linux发行版本和内核版本实验方法可能会有所不同。添加新的系统调用的步骤：

1. 下载一份内核源代码

Linux受GNU通用公共许可证（GPL）保护，其内核源代码是完全开放的。现在很多 Linux 的网站都提供内核代码的下载。推荐你使用 Linux 的官方网站：http://www.kernel.org。在这里你可以找到所有的内核版本。如：在www.kernel.org 或 http://mirrors.aliyun.com/linux-kernel/下载 4.8 版

本:linux-4.8.tar.xz 和 patch-4.8.xz（补丁）文件

2. 部署内核源代码

把内核代码文件 linux-4.8.tar.xz 存放在主目录（~）中。解压内核包，生成的内核源代码放在 linux.4.8 目录中：

```
tar -xvf linux-4.8.tar.xz
```

在 linux.4.8 目录中打内核补丁：

```
xz -d patch-4.8.xz | patch -p1
```

3. 配置内核

第 1 次编译内核的准备：

在 ubuntu 环境下，用命令 make menuconfig 对内核进行配置时，需要用终端模式下的字符菜单支持软件包 libncurses5-dev，因此你是第一次重建内核，需要下载并安装该软件包，下载并安装命令如下：

```
apt-get install libncurses5-dev libssl-dev
```

若上面这一步提示错误信息，则输入下面的命令 sudo apt-get -f install ，建立库依赖关系。

查看 README 文件：

在你进行这项工作之前，不妨先看一看~/linux 目录下内核源代码自带的 README 文件。在这份文件中，对怎样进行内核的解压，配置，安装都进行了详细的讲解。

配置内核：

在编译内核前，一般来说都需要对内核进行相应的配置。配置是精确控制新内核功能的机会。配置过程也控制哪些需编译到内核的二进制映像中(在启动时被载入)，哪些是需要时才装入的内核模块（module）。

```
cd ~/linux-4.8
```

第一次编译的话，有必要将内核源代码树置于一种完整和一致的状态。因此，我们推荐执行命令 make mrproper。它将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：

make mrproper

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：

cp /boot/config-`uname -r` .config

这里，命令`uname -r`得到当前内核版本号。然后：

make menuconfig

进行配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。例如，如果硬盘分区采用 ext2 文件系统（或 ext3 文件系统），则配置项应支持 ext2 文件系统（ext3 文件系统）。又例如，系统如果配有 SCSI 总线及设备，需要在配置中选择 SCSI 卡的支持。

对每一个配置选项，用户有三种选择，它们分别代表的含义如下：

- “<*>”或“[*]” — 将该功能编译进内核
- “[ ]” — 不将该功能编译进内核
- “[M]” — 将该功能编译成可以在需要时动态插入到内核中的模块

将与核心其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响。许多功能都可以这样处理，例如像上面提到的对 SCSI 卡的支持，等等。

4. 添加系统调用号

系统调用号在文件 unistd.h 里面定义。这个文件可能在你的 Linux 系统上会有两个版本：一个是 C 库文件版本，出现的地方是在 ubuntu 16.04 只修改/usr/include/asm-generic/unistd.h；另外还有一个版本是内核自己的 unistd.h，出现的地方是在你解压出来的内核代码的对应位置（比如 include/uapi/asm-generic/unistd.h）。当然，也有可能这个 C 库文件只是一个到对应内核文件的连接。现在，你要做的就是在文件 unistd.h 中添加我们的系统调用号：__NR_mysyscall，x86 体系架构的系统调用号 223 没有使用，我们新的系统调用号定义为 223 号，如下所示：

ubuntu 16.04 为：/usr/include/asm-generic/unistd.h

kernel 4.8 为：include/uapi/asm-generic/unistd.h

在/usr/include/asm-generic/unistd.h 文件中的查找定义 223 号的行，添

加或修改 223 号系统调用 mysyscall：

++ #define __NR_mysyscall 223

++ __SYSCALL(__NR_mysyscall, sys_mysyscall)

在文件 include/uapi/asm-generic/unistd.h 中做同样的修改

注意：不同版本的内核，需要修改路径（子目录名和文件名）可能不一样，根据实际版本号查找这些.h 文件。系统调用号也可能不一样，可以根据内核版本不同对系统调用号进行修改

添加系统调用号之后，系统才能根据这个号，作为索引，去找 syscall_table 中的相应表项。所以说，我们接下来的一步就是：

5. 在系统调用表中添加或修改相应表项

我们前面讲过，系统调用处理程序（system_call）会根据 eax 中的索引到系统调用表（sys_call_table）中去寻找相应的表项。所以，我们必须在那里添加我们自己的一个值。

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

223	common	mysyscall	sys_mysyscall

到现在为止，系统已经能够正确地找到并且调用 sys_mysyscall。剩下的就只有一件事情，那就是 sys_mysyscall 的实现。

6. 修改统计系统缺页次数和进程缺页次数的内核代码

由于每发生一次缺页都要进入缺页中断服务函数 do_page_fault 一次，所以可以认为执行该函数的次数就是系统发生缺页的次数。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。在当前进程控制块中定义一个变量 pf 记录当前进程缺页次数，在执行 do_page_fault 时，这个变量值加 1。

先在 include/linux/mm.h 文件中声明变量 pfcount：

++ extern unsigned long pfcount;

要记录进程产生的缺页次数，首先在进程 task_struct 中增加成员 pf01，在 include/linux/sched.h 文件中的 task_struct 结构中添加 pf 字段：

++ unsigned long pf;

统计当前进程缺页次数需要在创建进程是需要将进程控制块中的 pf 设置为 0，在进程创建过程中，子进程会把父进程的进程控制块复制一份，实现该复制过程的函数是 kernel/fork.c 文件中的 dup_task_struct()函数，修改该函数将子进程的 pf 设置成 0：

```
static struct task_struct *dup_task_struct(struct task_struct *orig)
{
    …..
    tsk = alloc_task_struct_node(node);
    if (!tsk)
        return NULL;
    ……
    ++ tsk->pf=0;
    ……
}
```

在 arch/x86/mm/fault.c 文件中定义变量 pfcount；并修改 arch/x86/mm/fault.c 中 do_page_fault()函数。每次产生缺页中断，do_page_fault()函数会被调用，pfcount 变量值递增 1,记录系统产生缺页次数，current->pf 值递增 1，记录当前进程产生缺页次数：

```
…
 ++ unsigned long pfcount;

 __do_page_fault(struct pt_regs *regs, unsigned long error_code)
{
     …
 ++ pfcount++;
 ++ current->pf++;
    …
}
```

7. sys_mysyscall 的实现

我们把这一小段程序添加在 kernel/sys.c 里面。在这里，我们没有在 kernel 目录下另外添加自己的一个文件，这样做的目的是为了简单，而且不用修改 Makefile，省去不必要的麻烦。

mysyscall 系统调用实现输出系统缺页次数、当前进程缺页次数，及每个进

程的“脏”页面数。

```
asmlinkage int sys_mysyscall(void)

{
      ……
//printk("当前进程缺页次数：%lu",current->pf)
//每个进程的“脏”页面数
     return 0;
}
```

8. 编译内核和重启内核

用 make 工具编译内核：

make

编译内核需要较长的时间，具体与机器的硬件条件及内核的配置等因素有关。完成后产生的内核文件 bzImage 的位置在~/linux/arch/i386/boot 目录下，当然这里假设用户的 CPU 是 Intel x86 型的，并且你将内核源代码放在~/linux 目录下。

如果编译过程中产生错误，你需要检查第 4、5、6、7 步修改的代码是否正确，修改后要再次使用 make 命令编译，直至编译成功。

如果选择了可加载模块，编译完内核后，要对选择的模块进行编译。用下面的命令编译模块并安装到标准的模块目录中：

sudo make modules

sudo make modules_install

通常，Linux 在系统引导后从/boot 目录下读取内核映像到内存中。因此我们如果想要使用自己编译的内核，就必须先将启动文件安装到/boot 目录下。安装内核命令：

sudo make install

我们已经编译了内核 bzImage，放到了指定位置/boot。现在，请你重启主机系统，期待编译过的 Linux 操作系统内核正常运行！

sudo reboot

9. 编写用户态程序

要测试新添加的系统调用，需要编写一个用户态测试程序（test.c）调用 mysyscall 系统调用。mysyscall 系统调用中 printk 函数输出的信息在 /var/log/messages 文件中（ubuntu 为 /var/log/kern.log 文件）。/var/log/messages (ubuntu 为/var/log/kern.log 文件)文件中的内容也可以在 shell 下用 dmesg 命令查看到。要求在 test.c 程序中输出 printk 函数输出的内容。

用户态程序

```
#include <linux/unistd.h>
# include <sys/syscall.h>
#define __NR_ mysyscall   223
int main()
{
      syscall(__NR_mysyscall);     /*或 syscall(223)  */
        //……输出相关的信息，要有较好的可视化
}
```

- 用 gcc 编译源程序

  gcc -o test test.c

- 运行程序

  ./test

完成实验后回答问题：

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？有什么区别？
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。
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。

撰写实验报告的要求

1. 按照下面实验报告模板格式撰写。
2. 整个实验过程的截图。
3. 源程序的修改部分，运行结果的截图。
4. 必须撰写实验讨论（即心得体会），内容为实验过程中遇到的问题及解决方法等。否则扣除本实验 20%分数。
5. 实验报告文件格式为 word 或 pdf，mysyscall()和 test.c 的源代码以文本形式附在实验报告所在的文件中，不要把 pdf 文件制作为图像格式，实验报告文件上传到“学在这里”中。

本实验评分参考：

1. 按时提交一个完整和规范的实验报告得 20 分：
2. 实验内容占 80 分：
   - 添加系统调用内核代码步骤占 10 分
   - 编译内核成功占 10 分
   - 编译测试程序成功及运行结果正确占 20 分
   - 讨论心得（实验过程中遇到的问题及解决方法）20 分
   - 实验问答题 20 分

浙江大学实验报告

课程名称：操作系统 实验类型：综合型

实验项目名称：

学生姓名：学号：

电子邮件地址：

实验日期：年 月 日

一、实验环境

填写您的计算机配置，操作系统环境，Linux 版本

二、实验内容和结果及分析

实验设计思路

实验步骤及截图

测试程序运行结果截图

结果分析

源程序

三、讨论、心得（20 分）

在这里写：实验过程中遇到的问题及解决的方法，您做本实验体会