lab08 New System Call

软件工程 2018 级 1813075 刘茵

一、 实验目标

- Add a new system call with arguments into the linux kernel
- The new system call will return all processes information to user mode

二、 操作过程

(一) PART 1

1. 进入 USR/SRC/LINUX 目录（和 HOME/LINUX5.8.15 相同）

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ cd include/linux/
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15/include/linux$ code syscalls.h
```

2. 用 vscode 打开 include/linux/syscall.h 文件，并修改

```
C sys.c    C syscalls.h ×
home > liuyin1813075 > linux-5.8.15 > include > linux > C syscalls.h > sys_alcall(int, char *)
1224  /*
1225   * Not a real system call, but a placeholder for syscalls which are
1226   * not implemented -- see kernel/sys_ni.c
1227   */
1228  asmlinkage long sys_ni_syscall(void);
1229
1230  asmlinkage long sys_schello(void);  /*lab07*/
1231
1232  asmlinkage long sys_alcall(int cmd,char* buf);/*lab08*/
1233  #endif /* CONFIG_ARCH_HAS_SYSCALL_WRAPPER */
1234
1235
```

3. 修改/kernel/sys.c 文件，添加 SYSCALL_DEFINE3 (alcall,int,cmd,char*,buf)函数

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ cd kernel/
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15/kernel$ code sys.c
```

构造了一个结构体，其中含有一个 buf2 数组和代表数组中进程数的 int 型变量 num；

利用 copy_to_user 和 put_user 两个函数实现内核空间数据与用户空间数据的相互访问。

```c
/*lab08 new systemcall*/
struct process{
    char buf2[1024];
    int num;
}
SYSCALL_DEFINE2(alcall, int, cmd, char *, buf)
{
    char temp[256];
    struct process allps;
    struct task_struct *p;
    snprintf(temp,256, "Hello new system call alcall (%d, %x)!\n", cmd, buf);
    printk("%s\n",temp);
    strcat(allps.buf2,temp);
    printk("LiuYin 1813075\n");
    snprintf(temp,256, "%-20s %-6s %-6s\n", "Name", "Pid", "Stat");
    printk("%s\n",temp);
    strcat(allps.buf2,temp);
    int count=0;
    for (p = &init_task; (p = next_task(p)) != &init_task;) {
        snprintf(temp, sizeof(temp), "%-20s %-6d %-6ld\n", p->comm,
            p->pid, p->state);
        printk("%s\n",temp);
        strcat(allps.buf2, temp);
        count++;
        if (count >= cmd)
            break;
    }
    allps.num=count;
    int ret=0;
    printk("Ready to copy to user");
    ret=copy_to_user(((struct process*)buf)->buf2,allps.buf2,strlen(allps.buf2));
    put_user(count,&(((struct process*)buf)->num));
    return ret;
}
```

4. 修改 arch/x86/entry/syscalls/syscall_64.tbl 文件，增加 441

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ cd arch/x86/entry/syscalls/
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15/arch/x86/entry/syscalls$ ls
arch  Makefile  syscall_32.tbl  syscall_64.tbl  syscallhdr.sh  syscalltbl.sh
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15/arch/x86/entry/syscalls$ code sys
call_64.tbl
```

syscall_64.tbl - Visual Studio Code

File Edit Selection View Go Run Terminal Help

syscall_64.tbl ×

home > liuyin1813075 > linux-5.8.15 > arch > x86 > entry > syscalls > syscall_64.tbl

```
431 common  fsconfig        sys_fsconfig
432 common  fsmount         sys_fsmount
433 common  fspick          sys_fspick
434 common  pidfd_open      sys_pidfd_open
435 common  clone3          sys_clone3
437 common  openat2         sys_openat2
438 common  pidfd_getfd     sys_pidfd_getfd
439 common  faccessat2      sys_faccessat2

440 common  schello         sys_schello
441 common  alcall          __x64_sys_alcall
#
```

5. re-configure the kernel

> cp linux_module .config
> make oldconfig
> make gconfig

6. 依次运行以下命令编译内核。

> make clean
> make -j5
> sudo make modules_install
> sudo make install

```
  OBJCOPY arch/x86/boot/setup.bin
  BUILD   arch/x86/boot/bzImage
Setup is 14140 bytes (padded to 14336 bytes).
System is 8937 kB
CRC 374401d9
Kernel: arch/x86/boot/bzImage is ready  (#5)
```

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ sudo make modules_install
[sudo] liuyin1813075 的密码:
  INSTALL drivers/thermal/intel/x86_pkg_temp_thermal.ko
  INSTALL fs/efivarfs/efivarfs.ko
  INSTALL net/ipv4/netfilter/iptable_nat.ko
  INSTALL net/ipv4/netfilter/nf_log_arp.ko
  INSTALL net/ipv4/netfilter/nf_log_ipv4.ko
  INSTALL net/ipv6/netfilter/nf_log_ipv6.ko
  INSTALL net/netfilter/nf_log_common.ko
  INSTALL net/netfilter/xt_LOG.ko
  INSTALL net/netfilter/xt_MASQUERADE.ko
  INSTALL net/netfilter/xt_addrtype.ko
  INSTALL net/netfilter/xt_mark.ko
  INSTALL net/netfilter/xt_nat.ko
  DEPMOD  5.8.15Lee202009
```

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ sudo make install
sh ./arch/x86/boot/install.sh 5.8.15Lee202009 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.8.15Lee202009 /boot/vmlinuz-5.8.15Lee
202009
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.8.15Lee202009 /boot/vmlinuz-5.8.15Lee2
02009
update-initramfs: Generating /boot/initrd.img-5.8.15Lee202009
find: '/var/tmp/mkinitramfs_eUEU4S/lib/modules/5.8.15Lee202009/kernel': 没有那个文件或目录
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.8.15Lee202009 /boot/vmlinuz-5.8.15
Lee202009
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.8.15Lee202009 /boot/vmlinuz-5.8.15Lee2
02009
run-parts: executing /etc/kernel/postinst.d/vboxadd 5.8.15Lee202009 /boot/vmlinuz-5.8.15Lee202009
VirtualBox Guest Additions: Building the modules for kernel 5.8.15Lee202009.

VirtualBox Guest Additions: Look at /var/log/vboxadd-setup.log to find out what
went wrong
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.8.15Lee202009 /boot/vmlinuz-5.8.15Lee20
2009
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
正在生成 grub 配置文件 ...
找到 Linux 镜像: /boot/vmlinuz-5.8.15Lee202009
找到 initrd 镜像: /boot/initrd.img-5.8.15Lee202009
找到 Linux 镜像: /boot/vmlinuz-5.8.15
找到 initrd 镜像: /boot/initrd.img-5.8.15
找到 Linux 镜像: /boot/vmlinuz-5.8.15.old
找到 initrd 镜像: /boot/initrd.img-5.8.15
找到 Linux 镜像: /boot/vmlinuz-5.4.0-54-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-54-generic
找到 Linux 镜像: /boot/vmlinuz-5.4.0-53-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-53-generic
找到 Linux 镜像: /boot/vmlinuz-5.4.0-42-generic
找到 initrd 镜像: /boot/initrd.img-5.4.0-42-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
```

7. 编写 testalcall.c 并运行。

syscall_64.tbl | testalcall.c | linux_module

home > liuyin1813075 > oscourse > course8 > testalcall.c > main(int, char * [])

```c
#include<unistd.h>
#include<sys/syscall.h>
#include<sys/types.h>
#include<stdio.h>
#define __NR_alcall 438
struct process{
    char buf2[1025];
    int num;
}
long alcall(int cmd,char *buf){
    return syscall(__NR_alcall,cmd,buf);
}
int main(int argc,char *argv[]){
    struct process result;
    int cmd;
    cmd=9;
    alcall(cmd,&result);
    printf("ok!run dmesg | grep alcall in terminal!%s\n",result,buf2);
    printf("the number of processes is %d\n",result.num);
    return 0;
}
```

编译并输出结果。

```
liuyin1813075@liuyin-VirtualBox:~/oscourse/course8$ ./testalcall
ok!run dmesg | grep alcall in terminal! Hello new system call alcall (9, 6b0f152
0)!
Name                 Pid     Stat
systemd              1       1
kthreadd             2       1
rcu_gp               3       1026
rcu_par_gp           4       1026
kworker/0:0H         6       1026
kworker/u8:0         7       1026
mm_percpu_wq         8       1026
ksoftirqd/0          9       1
rcu_sched            10      1026

the number of processes is 32765
```

系统调用输出。

```
liuyin1813075@liuyin-VirtualBox:~/oscourse/course8$ dmesg | grep alcall
[   99.101930] Hello new system call alcall (9, dcf16ef0)!
liuyin1813075@liuyin-VirtualBox:~/oscourse/course8$ dmesg
[    0.000000] Linux version 5.8.15Lee202009 (liuyin1813075@liuyin-VirtualBox) (
gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0, GNU ld (GNU Binutils for Ubuntu) 2.34)
 #5 SMP Fri Dec 4 16:59:52 CST 2020
```

```
[ 2028.626189] Hello new system call alcall (9, 3ae59b30)!

[ 2028.626193] LiuYin 1813075
[ 2028.626198] Name                 Pid    Stat

[ 2028.626202] systemd              1      1

[ 2028.626207] kthreadd             2      1

[ 2028.626211] rcu_gp               3      1026

[ 2028.626215] rcu_par_gp           4      1026

[ 2028.626219] kworker/0:0H         6      1026

[ 2028.626223] kworker/u8:0         7      1026

[ 2028.626227] mm_percpu_wq         8      1026

[ 2028.626231] ksoftirqd/0          9      1

[ 2028.626236] rcu_sched            10     1026
```