lab07 New System Call

软件工程 2018 级 1813075 刘茵

一、 实验目标

1. 向 LINUX 内核中添加新的系统调用

2. 在用户态下尝试新的系统调用

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:~$ cd /usr/src/linux
liuyin1813075@liuyin-VirtualBox:/usr/src/linux$ ls
arch          fs          LICENSES                net          usr
block         include     MAINTAINERS             README       virt
certs         init        Makefile                samples      vmlinux
COPYING       ipc         mm                      scripts      vmlinux-gdb.py
CREDITS       Kbuild      modules.builtin         security     vmlinux.o
crypto        Kconfig     modules.builtin.modinfo sound        vmlinux.symvers
Documentation kernel      modules.order           System.map
drivers       lib         Module.symvers          tools
```

2. 用 vscode 打开 include/linux/syscall.h 文件，并修改。增加 1230 行。

```
liuyin1813075@liuyin-VirtualBox:/usr/src/linux$ code ./include/linux/syscalls.h
```

home > liuyin1813075 > linux-5.8.15 > include > linux > C syscalls.h > sys_schello(void)

```
1202  /* obsolete: fs/readdir.c */
1203  asmlinkage long sys_old_readdir(unsigned int, struct old_linux_dirent __user *
1204
1205  /* obsolete: kernel/sys.c */
1206  asmlinkage long sys_gethostname(char __user *name, int len);
1207  asmlinkage long sys_uname(struct old_utsname __user *);
1208  asmlinkage long sys_olduname(struct oldold_utsname __user *);
1209  #ifdef __ARCH_WANT_SYS_OLD_GETRLIMIT
1210  asmlinkage long sys_old_getrlimit(unsigned int resource, struct rlimit __user
1211  #endif
1212
1213  /* obsolete: ipc */
1214  asmlinkage long sys_ipc(unsigned int call, int first, unsigned long second,
1215          unsigned long third, void __user *ptr, long fifth);
1216
1217  /* obsolete: mm/ */
1218  asmlinkage long sys_mmap_pgoff(unsigned long addr, unsigned long len,
1219                 unsigned long prot, unsigned long flags,
1220                 unsigned long fd, unsigned long pgoff);
1221  asmlinkage long sys_old_mmap(struct mmap_arg_struct __user *arg);
1222
1223
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);
1231  #endif /* CONFIG_ARCH_HAS_SYSCALL_WRAPPER */
1232
1233
```

3. 修改/kernel/sys.c 文件，添加 SYSCALL_DEFINE0(schello)函数

```
liuyin1813075@liuyin-VirtualBox:/usr/src/linux/kernel$ code ./sys.c
```

home > liuyin1813075 > linux-5.8.15 > kernel > C sys.c > ...

```
904    */
905   SYSCALL_DEFINE0(getpid)
906   {
907       return task_tgid_vnr(current);
908   }
909
910   /* Thread ID - the internal kernel "pid" */
911   SYSCALL_DEFINE0(gettid)
912   {
913       return task_pid_vnr(current);
914   }
915
916   SYSCALL_DEFINE0(schello)
917   {
918    printk("Hello new system call schello!\n");
919    return 0;
920   }
```

4. 修改 arch/x86/entry/syscalls/syscall_64.tbl 文件，增加 440

```
liuyin1813075@liuyin-VirtualBox:/usr/src/linux/kernel$ cd ..
liuyin1813075@liuyin-VirtualBox:/usr/src/linux$ cd arch/x86/entry/syscalls/
liuyin1813075@liuyin-VirtualBox:/usr/src/linux/arch/x86/entry/syscalls$ ls
Makefile  syscall_32.tbl  syscall_64.tbl  syscallhdr.sh  syscalltbl.sh
liuyin1813075@liuyin-VirtualBox:/usr/src/linux/arch/x86/entry/syscalls$ pwd
liuyin1813075@liuyin-VirtualBox:/usr/src/linux/arch/x86/entry/syscalls$ code syscall_64.tbl
```

syscall_64.tbl ×

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

```
425 common  io_uring_setup      sys_io_uring_setup
426 common  io_uring_enter      sys_io_uring_enter
427 common  io_uring_register   sys_io_uring_register
428 common  open_tree       sys_open_tree
429 common  move_mount      sys_move_mount
430 common  fsopen          sys_fsopen
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

#
# x32-specific system call numbers start at 512 to avoid cache
# for native 64-bit operation. The __x32_compat_sys stubs are
# on-the-fly for compat_sys_*() compatibility system calls if
# is defined.
#
```

5. 依次运行以下命令编译内核。

```
make clean
make -j5
sudo make modules_install
sudo make install
```

```
liuyin1813075@liuyin-VirtualBox:/usr/src/linux$ make clean
  CLEAN   arch/x86/crypto
  CLEAN   arch/x86/entry/vdso
  CLEAN   arch/x86/kernel/cpu
  CLEAN   arch/x86/kernel
  CLEAN   arch/x86/purgatory
  CLEAN   arch/x86/realmode/rm
  CLEAN   arch/x86/lib
  CLEAN   certs
  CLEAN   crypto/asymmetric_keys
  CLEAN   drivers/eisa
  CLEAN   drivers/firmware/efi/libstub
  CLEAN   drivers/gpu/drm/radeon
```

```
Linux liuyin-VirtualBox 5.8.15 #1 SMP Sat Nov 28 03:12:15 CST 2020 x86_64 x86
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ sudo make modules_install
  INSTALL arch/x86/crypto/aegis128-aesni.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
  INSTALL arch/x86/crypto/blowfish-x86_64.ko
```

```
  INSTALL sound/xen/snd_xen_front.ko
  DEPMOD  5.8.15
depmod: ERROR: failed to load symbols from /lib/modules/5.8.15/misc/vboxsf.ko: Invalid argument
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$
```

Ps: make modules_install 结束后会报错 error(但是对结果没有影响)，网上没有找到具体的匹配信息。

```
liuyin1813075@liuyin-VirtualBox:~/linux-5.8.15$ sudo make install
sh ./arch/x86/boot/install.sh 5.8.15 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.8.15 /boot/vmlinuz-5.8.15
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.8.15 /boot/vmlinuz-5.8.15
update-initramfs: Generating /boot/initrd.img-5.8.15

run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.8.15 /boot/vmlinuz-5.8.15
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.8.15 /boot/vmlinuz-5.8.15
run-parts: executing /etc/kernel/postinst.d/vboxadd 5.8.15 /boot/vmlinuz-5.8.15
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.8.15 /boot/vmlinuz-5.8.15
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-5.8.15
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
```

6. 编写 testschello.c 程序在用户态测试系统调用，并执行

> gcc -o testschello testschello.c
> ./testschello
> dmesg | grep schello （查看输出）

```
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ vim testschello.c
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ ls
testschello.c
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ cat testschello.c
#include <unistd.h>
#include <sys/syscall.h>
#include <sys/types.h>
#include <stdio.h>
#define __NR_schello 440
int main(int argc, char *argv[])
{
syscall(__NR_schello);
printf("ok! run dmesg | grep hello in terminal!\n");
return 0;
}
```

```
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ gcc  -o testschellot testsch
ello.c
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ ./testschello
ok! run dmesg | grep hello in terminal!
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ dmesg | grep hello
[  275.751226] Hello new system call schello!
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ dmesg | grep schello
[  275.751226] Hello new system call schello!
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$
```

(二) Part 2

1. 再次在 kernel/sys.c文件中修改函数定义输出当前所有进程：

```
SYSCALL_DEFINE0(schello)
{
     struct task_struct *p;
printk("Hello new system call schello!\n");
printk("%-20s %-6s %-6s\n","Name","Pid","Stat");
for (p = &init_task; (p = next_task(p)) != &init_task;)
printk("%-20s %-6d %-6ld\n",p->comm,p->pid,p->state);
return 0;
}
```

2. 重复<Part 1>的 5-7 步骤
3. 再次输出结果。

```
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ gcc  -o testschellot testsch
ello.c
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ ./testschello
ok! run dmesg | grep hello in terminal!
liuyin1813075@liuyin-VirtualBox:~/oscourse/course7$ dmesg | grep hello
[   62.137581] Hello new system call schello!
[   62.137905] testschello          1973   0
```