New System Call

Under Linux Kernel 4/5.x

Linux Kernel

- www.kernel.org

  linux-4.13.6.tar.xz

- uname -a

  Linux ubuntu 4.13.6 #2 SMP Tue Oct 24 22:36:32 PDT 2017 i686 i686 i686 GNU/Linux

Add New System Call - schello

- Step 1)
- include/linux/syscalls.h
- 在文件

  #endif /* CONFIG_ARCH_HAS_SYSCALL_WRAPPER */

  之前，添加一行

  asmlinkage long sys_schello(void);

Add New System Call - schello

- Step 2)
- kernel/sys.c
- ? 在文件 SYSCALL_DEFINE0(gettid) 函数之后，添加如下行

```
SYSCALL_DEFINE0(schello)
{
    printk("Hello new system call schello!\n");
    return 0;
}
```

Add New System Call - schello

- Step 3a)
  - arch/x86/entry/syscalls/syscall_32.tbl
  - 在文件 384 i386 arch_prctl sys_arch_prctl compat_sys_arch_prct

  行之后，添加如下行

  385 i386 schello sys_schello
- Step 3b)
  - arch/x86/entry/syscalls/syscall_64.tbl
  - 在文件 334 common rseq __x64_sys_rseq
- 行之后，添加如下行
- 335 common schello __x64_sys_schello

Add New System Call - schello

- Step 4)
- 重新编译内核

  make clean

  make -j5

  sudo make modules_install

  sudo make install

Add New System Call - schello

- Step 5)
- 编写用户态测试程序testschello.c

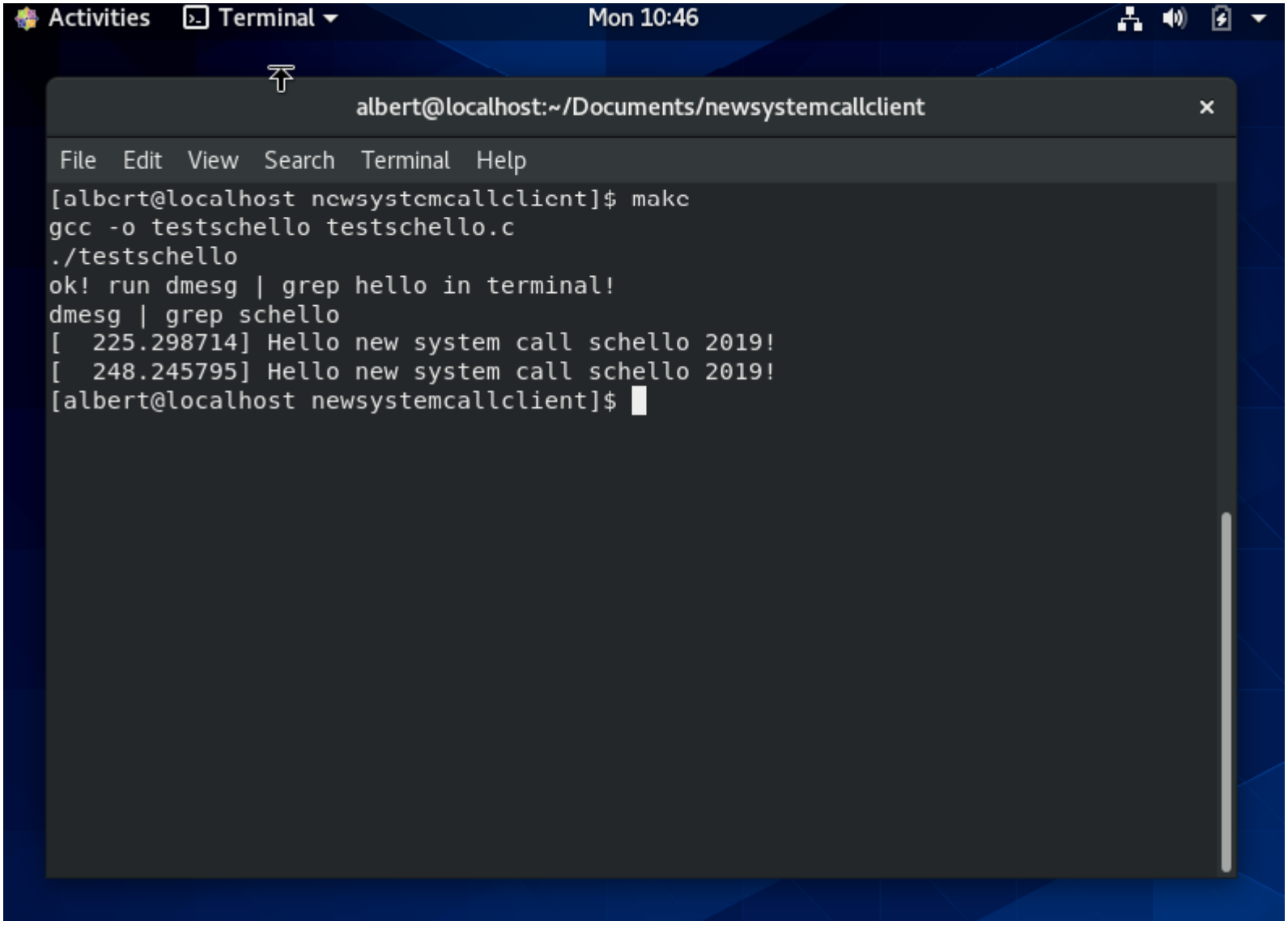
```
#include <unistd.h>
#include <sys/syscall.h>
#include <sys/types.h>
#include <stdio.h>
#define __NR_schello 385
int main(int argc, char *argv[])
{
syscall(__NR_schello);
printf("ok! run dmesg | grep hello in terminal!\n");
return 0;
}
```

Add New System Call - schello

- Step 6)
- 编译用户态测试程序 testschello.c， 并执行

```
gcc  -o testschello   testschello.c
./testschello
$dmesg | grep schello
[ 1648.215250] Hello new system call schello!
```

Add New System Call - schello



Enhance New System Call - schello

- Step 2)
- kernel/sys.c
- 在文件SYSCALL_DEFINE0(gettid)函数之后，添加如下行

```
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;

}
```

End