

New System Call

Under Linux Kernel 4/5.x


Linux Kernel

File Edit View History Bookmarks Tools Help

The Linux Kernel Archives

https://www.kernel.org


The Linux Kernel Archives



[About](#) [Contact us](#) [FAQ](#) [Releases](#) [Signatures](#) [Site news](#)

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

5.9.8 

mainline:	5.10-rc4	2020-11-16	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]		
stable:	5.9.8	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	5.8.18 [EOL]	2020-11-01	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.77	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.157	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.206	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.243	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.243	2020-11-10	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20201116	2020-11-16						[browse]	

Linux Kernel

- `www.kernel.org`
`linux-5.8.7.tar.xz`
- `uname -a`

```
Linux cov-VirtualBox 5.8.7Lee202009 #3 SMP  
Fri Oct 30 05:10:10 EDT 2020 x86_64 x86_64  
x86_64 GNU/Linux
```

Add New System Call - schello

- Step 1)
- include/linux/syscalls.h
- 在文件 (No. 1230)

```
#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 (No. 911) , 添加如下行

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

Add New System Call - schello

- 针对 64 位 OS
- Step 3b)
 - **arch/x86/entry/syscalls/syscall_64.tbl**
 - (4.xkernel) ~~在文件 334 common rseq~~ ~~_____x64_sys_rseq~~
 - 在文件 439 common faccessat2 sys_faccessat2
- 行之后, 添加如下行
- (4.xkernel)~~335 common schello~~
~~_____x64_sys_schello~~
- 440 common schello 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 335
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

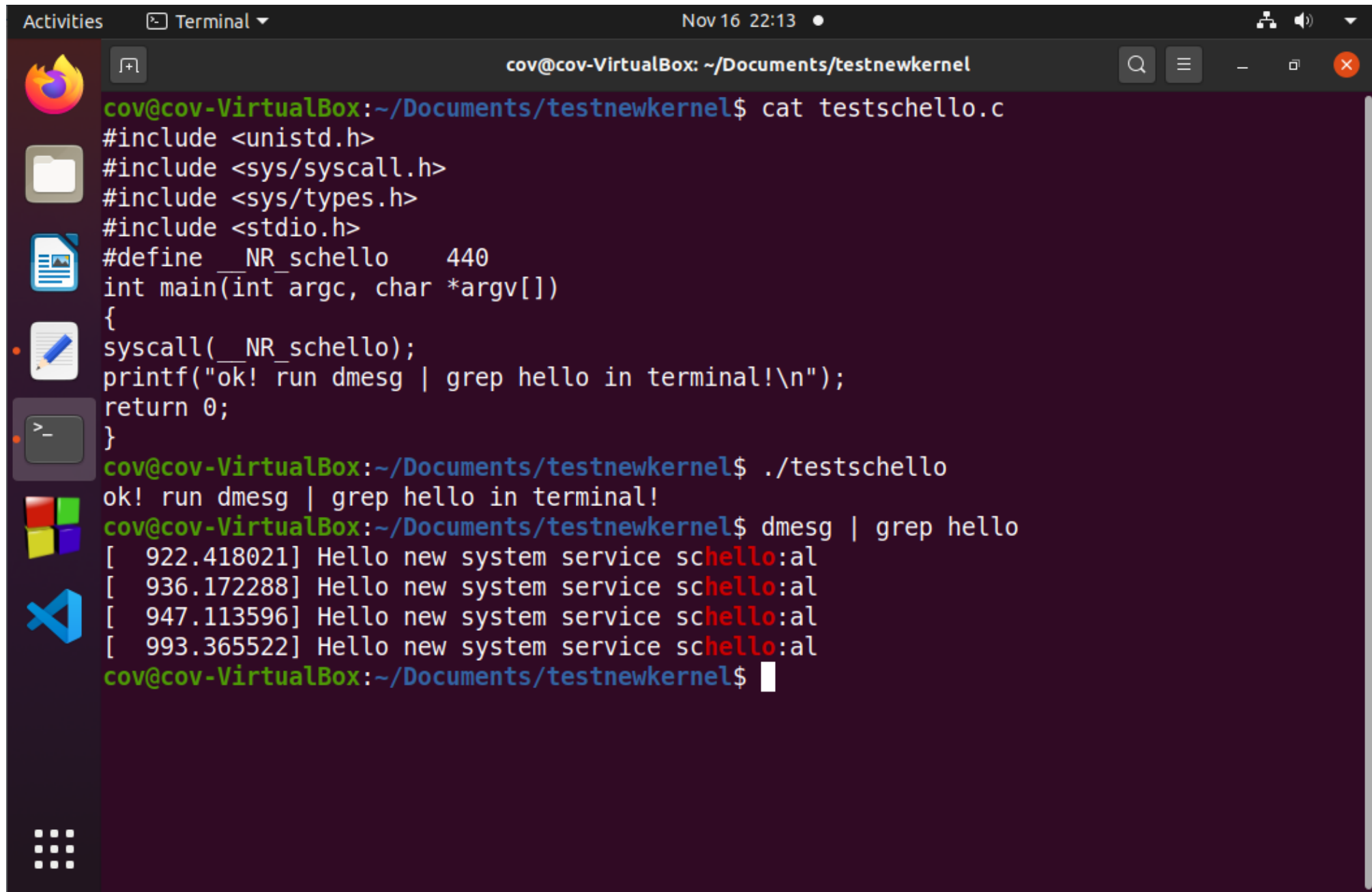
```
gcc -o testschello testschello.c
```

```
./testschello
```

```
$dmesg | grep schello
```

```
[ 1648.215250] Hello new system call schello!
```

Add New System Call - schello



The screenshot shows a terminal window titled "cov@cov-VirtualBox: ~/Documents/testnewkernel". The user has created a file named `testschello.c` with the following content:

```
#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;
}
```

The user then runs the program with `./testschello`, which outputs "ok! run dmesg | grep hello in terminal!". Finally, the user runs `dmesg | grep hello`, which displays four log entries from the kernel:

```
[ 922.418021] Hello new system service schello:al
[ 936.172288] Hello new system service schello:al
[ 947.113596] Hello new system service schello:al
[ 993.365522] Hello new system service schello:al
```

The terminal window includes a sidebar with application icons (Firefox, Files, LibreOffice, etc.) and a top bar with system information (date, time, and window controls).

End