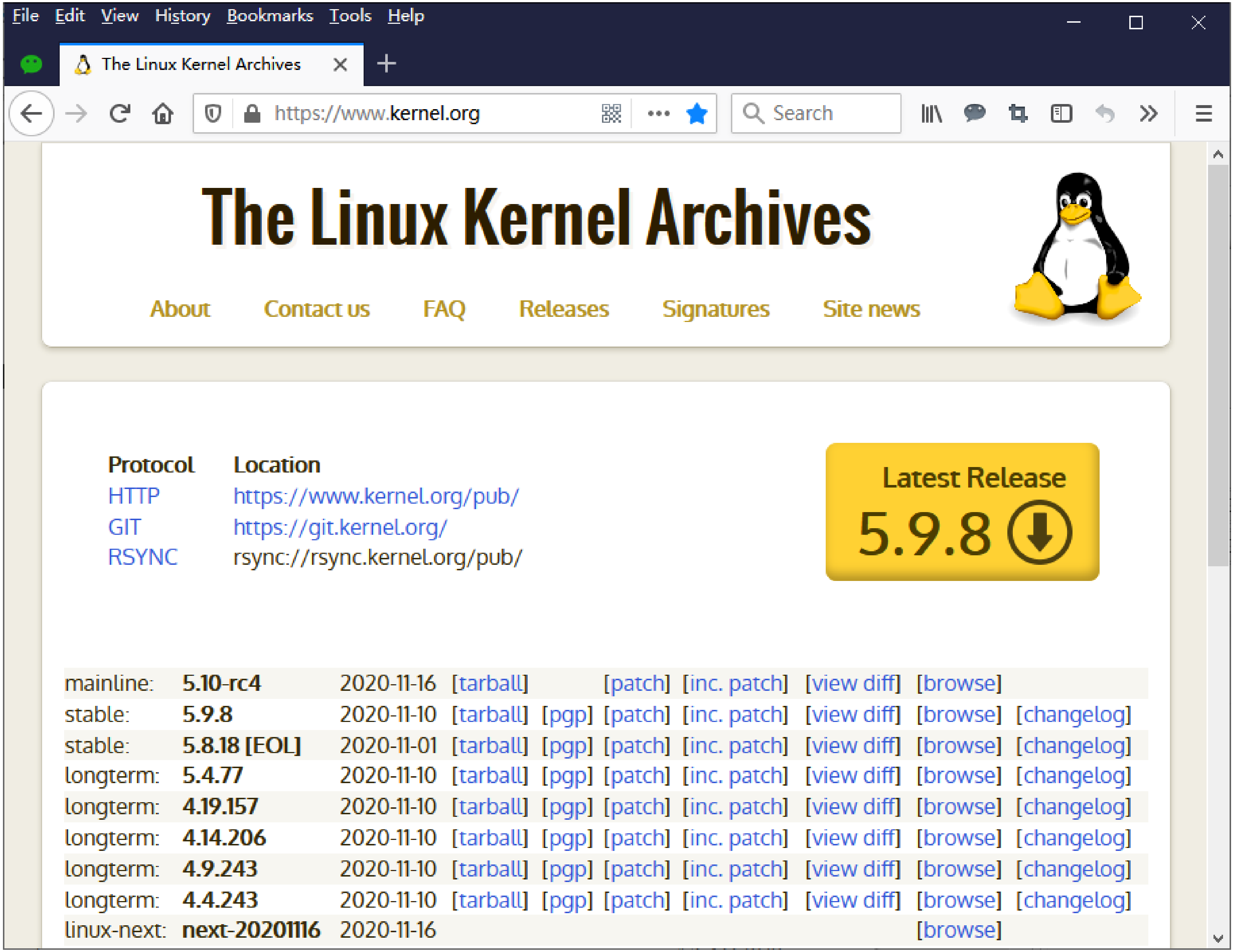
New System Call

Under Linux Kernel 4/5.x

Linux Kernel



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

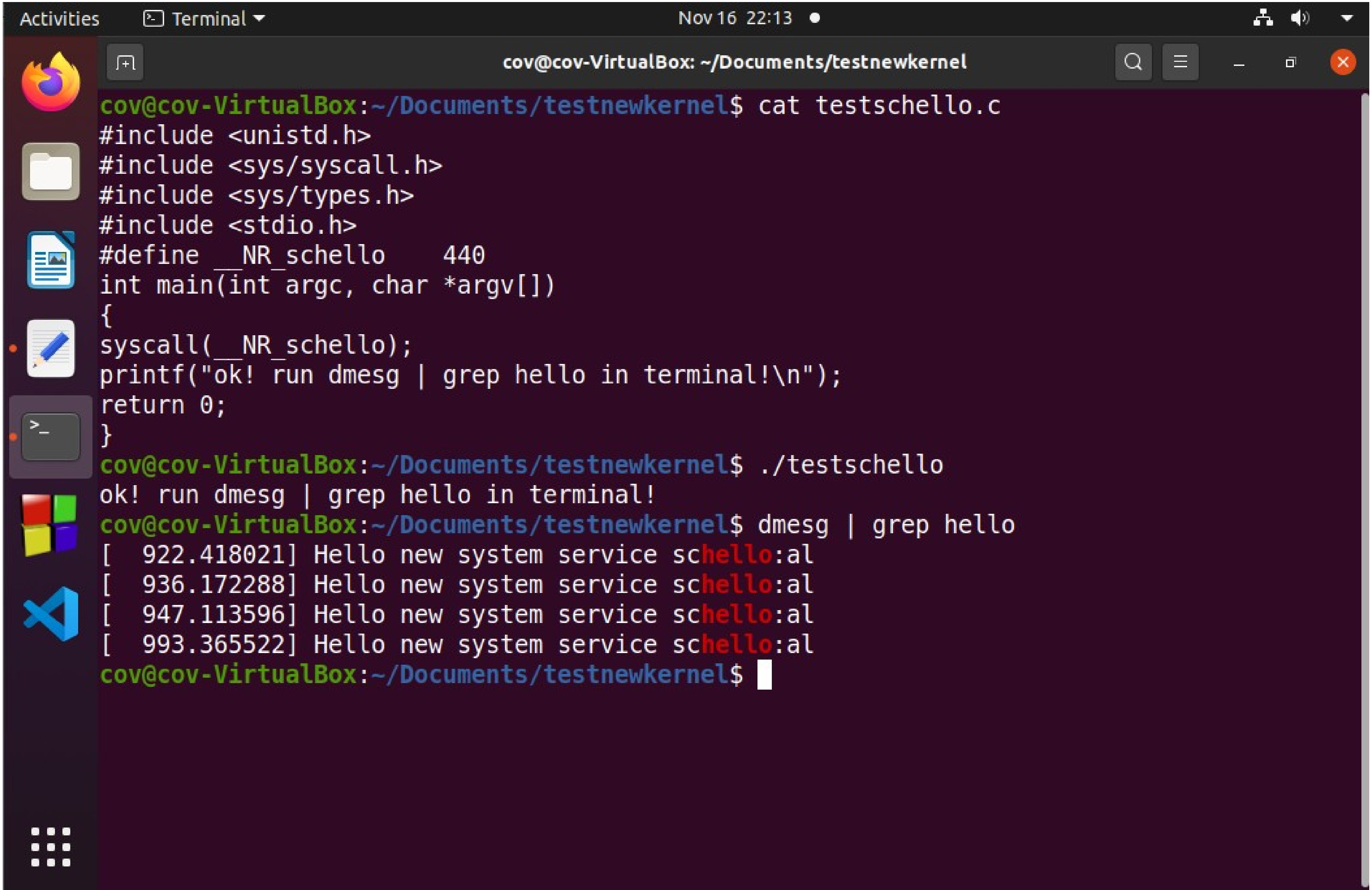
```
#include <unistd.h>
#include <sys/syscall.h>
#include <sys/types.h>
#include <stdio.h>
#define __NR_schello   335
int main(int argc, char *argv[])
{
syscall(_NR_schello);
printf("ok! run dmesg | grep hello in terminal!\n");
return 0;
}
```

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



```
cov@cov-VirtualBox:~/Documents/testnewkernel$ 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;
}
cov@cov-VirtualBox:~/Documents/testnewkernel$ ./testschello
ok! run dmesg | grep hello in terminal!
cov@cov-VirtualBox:~/Documents/testnewkernel$ dmesg | grep hello
[  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
cov@cov-VirtualBox:~/Documents/testnewkernel$ 
```

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