

CSIS0230A Principle of Operating Systems (Class A)

Tutorial 4

Making a new Linux system call

We will modify the kernel ourselves to make a new system call. To start with, our system call will be trivial to implement. Our focus is the process needed to create the system call.

From <http://www.csis.hku.hk/~c0230a/t03/>, get the 2.2.19 kernel. Extract files in it, but don't move the resulting `linux` directory to `/usr/src`. Reuse the 686 config of the 2.2.14 kernel as usual. Then

(0) **Compile the kernel** once (i.e., `make bzImage`), after following Hints 1 and 2 below. (1) **Modify the kernel** according to the specifications below. Then compile it. You may find Hints 3–6 useful. (2) Install the new kernel image at `/boot/vmlinuz-exp`, modify `lilo.conf` accordingly, and **boot the new kernel**. *Don't make the new kernel the default*, as the kernel is experimental. You don't need to move the `System.map` file to `/boot`. (3) **Write a user program that calls your new system calls**, in the sequence `getcount`, `deccount`, `inccount` and `deccount`.

Specifications: Add an initially 0 counter to the kernel, which can hold a non-negative 16-bit counter (i.e., type `short`), with value from 0 to 32767. There should be three system calls to manipulate it: `sys_inccount` increment the count, `sys_deccount` decrement it, and `sys_getcount` get the count. All attempts to make the counter out of range should be denied, generating an error `EINVAL` (Invalid argument). On the other hand, successful changes to the counter should also return the new value of the counter. The new system calls should be implemented in a new file `kernel/counter.c` of the kernel source tree.

Hint 0: You can do everything as **root** in this tutorial. Password is **tutorial**.

Hint 1: You probably want to skip the unneeded modules. To do this, edit the `.config` file to turn all `=m` (compile as module) things to `=n` (not compiled). In Emacs, you can use `M-x replace-string` for mass replacement.

Then type `make menuconfig` and go to “Network devices support”, “Ethernet (10 or 100Mbps)”, and say ‘y’ to 3c590 support there. Then go back to the main menu, go to “Networking options”, and say ‘n’ to “IP-masquarading”. Now exit, saving the configuration.

Hint 2: To speed up compilation, you may remove these options in `make menuconfig`:

- **Top-level:** *SCSI support*.
- **Processor type and features:** *Maths emulation, MTRR support*.

Hint 3: Include the `<linux/kernel.h>` header! The `asmlinkage` macro is defined there.

Hint 4: It is usually beneficial to *printk* a few things here and there for debugging.

Hint 5: Don't forget to modify `entry.S` and `Makefile`.

Hint 6: Even though you won't use modules, type `make modules_install` anyway (or make the directory `/lib/modules/2.2.19`). Otherwise Redhat boot scripts will complain.

Now is time to get your hands dirty.