Project II: System Call and Synchronization Primitive
Deadline: 11:59 PM, April 29 2011.

In this homework assignment, you will learn how to modify and compile a new FreeBSD kernel as well as how to utilize the ”kld” (Kernel Loadable Module) interface to change the behavior of the kernel such as adding new system calls. Information regarding how to compile a FreeBSD kernel and how to program the KLD is available in the class website. During the discussion hours, the TA will also talk about kernel compilation and dynamically loadable kernel modules. The TA will provide a test program as well.

You must use the 5.4 kernel to do this and other kernel programming assignments, and you need to submit all the kernel source code files you modify as well as the Makefile. Please do NOT hand-in the whole source tree or any binary. You also need to write a README file regarding which files you have changed, and a brief description (less than two pages) about the steps for a system call to trap into the FreeBSD kernel. In order to do this part, you need to identify the path of ”system call” implementation in the FreeBSD source tree.

In this programming assignment, you will add one new attribute ”int tickets” into ”struct proc” defined in ”`/usr/src/sys/sys/proc.h`”, and one global variable ”`int lottery_mode`” in ”`/usr/src/sys/kern/kern_switch.c`”. Then, you will add the following four system calls via the KLD interface:

```
int setProcessTickets(int pid, int tickets);
int getProcessTickets(int pid);
int setLotteryMode(int mode);
int getLotteryMode(void);
```

The first two system calls will return an error code (-1) if the PID doesn’t exist in the kernel. Please note that you need to re-compile the kernel in order to add the new attribute (tickets in `struct proc`) and the new global variable (`lottery_mode`), but you don’t need to re-build the kernel to add new system calls.

Finally, in the development of this programming assignment, you will likely use the `printf` function call. Please also describe, in the README file, the difference between calling `printf` in the user space and in kernel space (i.e., in the kernel module).

If you encounter any difficulty, please feel free to ask the instructor or the TA.

Students can form groups (of at most two) to solve this problem.