

---

**Project III: Lottery Scheduling in the FreeBSD 5.x Kernel**  
**Deadline: 11:59 PM; 5/17/2011**

This assignment is related to "lottery scheduling" algorithm in FreeBSD 5.4 kernel. Please make sure that you follow a good coding convention. You need to provide a Makefile and a README.txt.

Modify the FreeBSD 5.x kernel such that the users can give different amount of lottery tickets to each "user" process. Please note that you do not need to worry about "lottery ticket sharing" case - just the simplest form of lottery scheduling. We still maintain different levels of priorities as in FreeBSD, and the lottery mechanism will be placed ONLY in the time-sharing scheduling class (i.e., priorities 160 223).

In programming assignment II, you have added 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". And, you have added the following four system calls:

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

Now, we will complete the implementation of lottery scheduling in this project.

First, by default, each user process in the time sharing class has FOUR tickets, while a user program can set the ticket number differently using the `setProcessTickets` system call. Second, your kernel uses "normal BSD scheduling" by default, and the amount of tickets in each process makes no impact to the scheduling behavior. However, when a user process sets the `lottery_mode` to 1 (via the `setLotteryMode` system call), the kernel changes its scheduling behavior immediately to "lottery scheduling".

The original Lottery Scheduling can be found at:

[http://www.usenix.org/publications/library/proceedings/osdi/full\\_papers/waldspurger.pdf](http://www.usenix.org/publications/library/proceedings/osdi/full_papers/waldspurger.pdf)

Your team (of two) will use the modified version of "top" (details forthcoming) to test whether your implementation distributes CPU cycles to the process according to the lottery scheduling principle. Furthermore, to simplify the task, we assume that all KSEs (or kernel threads) will obtain exactly the same number of tickets being assigned to its process. You don't need to implement lottery ticket sharing.

Among possibly others, you will likely modify a few kernel source files such as:

```
kern/kern_switch.c
kern/kern_fork.c
```

Also, you will need to use the file `sys/proc.h` from Project II. The global variable `lottery_mode` indicates whether the lottery scheduler is on or not (0: off, 1: on).