← Project 5: A kernel module

Due by midnight, Sunday 12/9 (or late on Monday)

In this project, you will write a kernel module to create a new software device, `/dev/dice`, which returns randomly selected rolls of a 6-sided die. You'll also write a very simple game that can make use of that kernel module.

This project will give you practice with using a **virtual machine** - a kind of emulator for running an operating system inside another operating system.

Grading Breakdown

- **[5]** Submitted properly
  - See instructions below!!!
- **Craps [55]**
  - **[8]** compiles with `gcc -Wall -Werror --std=c99 -m32 -static -o craps craps.c`
  - **[10]** reads random numbers from the file specified on the command line (instead of `rand()`)
  - **[10]** checks for errors when doing `open/read/fopen/fread` calls
  - **[10]** asks for username and asks for play/quit using **strings,** not numbers
  - **[10]** properly implements rules of the game
  - **[5]** code style
- **Kernel module [40]**
  - **[5]** compiles with `make ARCH=i386`
  - **[10]** properly updates `ppos`
  - **[10]** generates random numbers
    - *without* making a large stack buffer like `unsigned char rolls[count];`
  - **[10]** handles an arbitrary number of bytes for the output
  - **[5]** code style

Submission Instructions

> Remove or comment out all debugging `printf/printk` statements. Put your name at the top of each source file. Make a copy of your code somewhere.

Create a tar archive (tarchive?) containing two things:

- The entire folder containing your kernel module's **source code, Makefile, and rules**
  - The grader should be able to `cd` into it and `make ARCH=i386` and see that it builds.
- Your `craps.c` file.

Now you can **submit as usual.**